uc3m | Universidad Carlos III de Madrid

Grado en Ingeniería en Tecnologías

de Telecomunicación

2016-2017

Trabajo Fin de Grado

Implementación y análisis de mecanismos de autenticación basados en eye tracking

---

Sergio Cerrada Lerena

Tutor/es

Florina Almenarez Mendoza

Patricia Arias Cabarcos

Agradecimientos

En primer lugar, me gustaría agradecer todo lo que mi familia y pareja han hecho por mí, todos y cada uno de ellos han sido un pilar fundamental en mi etapa universitaria, dándome siempre el apoyo y la motivación que he necesitado.

En segundo lugar, quiero dar las gracias a Patricia y Florina, por haber confiado en mí para realizar este trabajo fin de grado y por todo lo que me han enseñado durante mi etapa universitaria.

Por último, quiero agradecer a todos mis compañeros, en especial a Sergio, Ángel, Chacón y Edgar, ya que hemos tenido buenos y malos momentos, pero siempre nos hemos mantenido juntos y gracias a ellos hemos podido con todo.

Resumen

La autenticación es la capacidad de demostrar que un usuario es realmente quién asegura ser, y es uno de los procesos principales que suele utilizarse como control de acceso a los recursos utilizados en un sistema. Tradicionalmente han existido como mecanismos de autenticación el uso de credenciales, usuario y contraseña, para Internet y código PIN o patrón táctil para dispositivos móviles. Estos mecanismos de autenticación presentan evidentes vulnerabilidades, como el ataque conocido como *Shoulder Surfing*, el cual consiste en observar directamente el dispositivo de otra persona con el fin de obtener cualquier información útil.

La seguridad de los datos personales que se almacenan en los dispositivos electrónicos es una de las mayores preocupaciones de los usuarios en la actualidad. Además de la búsqueda de mayor seguridad en los dispositivos electrónicos, los usuarios valoran positivamente la usabilidad de los mecanismos de autenticación, es decir, que sean cómodos, fáciles de manejar y rápidos.

Optimizar el proceso de autenticación ha sido uno de los objetivos prioritarios durante los últimos años para los investigadores y compañías. Aunque la cantidad de sistemas en los que nos autenticamos ha aumentado, los mecanismos de autenticación siguen siendo los tradicionales. Observando que este ámbito está en continuo crecimiento y sabiendo que es algo que interesa a millones de usuarios que valoran la integridad de su información, se va a realizar el presente Trabajo Fin de Grado.

Durante éste se van a implementar y comparar diversos mecanismos de autenticación basados en el reconocimiento y seguimiento de la mirada mediante el dispositivo *The Eye Tribe*. **Estos tipos de autenticación** son interesantes ya que tienen el potencial de aumentar la usabilidad de los mecanismos de autenticación, serían útiles para usuarios con discapacidad o escenarios donde introducir la contraseña pueda resultar incómodo, y a la vez reducen el riesgo de sufrir ataques, como el mencionado *shoulder surfing*.

En las siguientes páginas de este documento, se va a detallar la implementación de cada uno de los mecanismos de autenticación y se van a desglosar los resultados de las pruebas finales.

Palabras clave: seguridad, mecanismo de autenticación, vulnerabilidades, *The Eye Tribe*, reconocimiento y seguimiento de la mirada.

Abstract

Authentication is the ability to demonstrate that a user is really who he or she claims to be, and is one of the main processes that is often used as access control to the resources used in a system. The most used authentication mechanisms in the past were the use of credentials, user and password, for the Internet and PIN code access or touch pattern for mobile devices. These authentication mechanisms have several vulnerabilities such as the Shoulder Surfing attack which consists on observing directly into another person's device in order to obtain any valuable information.

The security of the personal stored data in electronic devices is currently one of the main concerns of the users. Furthermore, the users see as a positive factor the good usability of the authentication mechanisms, meaning comfortable, easy to handle and fast authentication mechanisms.

Optimizing the authentication process has been one of the main goals over the last few years for researchers and companies. Despite the amount of systems in which we authenticate has increased, the traditional authentication mechanisms are still being the most used. Regarding that this area is constantly growing and knowing that this matter interests to millions of users that appreciate the integrity of their information, this bachelor thesis is going to be carried out.

Along this project, various authentication mechanisms based on the eye tracking technology with The Eye Tribe device are going to be implemented and compared. **This kind of authentication mechanisms** are interesting as they have the potential to increase the usability of the authentication mechanisms, it is useful for people with disabilities or for scenarios where entering the password is uncomfortable, and at the same time they reduce the risk of suffering attacks, such as the mentioned before shoulder surfing attack.

In the following pages of this document, the implementation of each of the authentication mechanisms and their test results will be detailed.

Keywords: security, authentication mechanisms, vulnerabilities, The Eye Tribe, eye tracking.

Índice

Índice de figuras

Índice de tablas

Glosario

API → Application Programming Interface
CSS → Cascading Stylesheets
IDE → Integrated Development Environment
JDK → Java Development Kit
JRE → Java Runtime Environment
LOPD → Ley Orgánica de Protección de Datos
MFA → Multi-factor Authentication
NIST → National Institute of Standards and Technology
OS → Operating System
PIN → Personal Identification Key
PIT → Personal Identification Text
SDK → Software Development Kit
SMI → SensoMotoric Instruments
TPS → Theoretical Password Spaces
UI → User Interface

Capítulo 1

Introduction

It is a known fact that the number of users that own an electronic device increases each year considerably. Also, the trend is to store more and more personal information on devices such as passwords, contacts, appointments or images, which means a greater amount of useful data for attackers.

This is why device security is of the utmost importance. But security is composed of several levels and in this case, we are going to deal with the most visible level for the user, the authentication to access into the device.

1.1 Goals and motivation

The most used authentication mechanisms during the last years, passcode-based and finger movement pattern-based, do not adapt to the emerging needs as these mechanisms are vulnerable to different known attacks. Users are increasingly concerned about the security of their authentication, wanting to protect all the personal data stored on their devices.

Data protection is a very important issue in manufacturing companies nowadays, as the number of devices and systems that require authentication continues to grow. In addition to this, it is necessary that the authentication mechanisms are as usable as possible because the users prefer systems that are not complicated and systems that require the minimal effort to use and understand. We live in a digital age and it is crucial that the security of the devices in which we store all the information is as strong as possible while the system is easy to use.

The question arising here is *how can the authentication mechanisms be safer and more usable?* There are two main paths followed by the researchers and companies in order to achieve a more secure authentication for the users:

- **Multi-factor authentication (MFA)**, which is a mechanism where the user can only access to the device once he can provide two or more different proofs that he is the owner of the device. These proofs can be a password, a secondary password or a digital certificate for instance.

- **Biometrics-based** authentication, which consists on using morphological characteristics of the user to decide if he can either access or not to the device. In relation to this, during this project, different authentication mechanisms are going to be implemented based on the gaze of the eyes.

As far as I know, there are already company departments[1] that are investing in the development of authentication mechanisms based on the eyetracking technology. This is a clear sign that authenticating using advanced mechanisms is a reality and will soon leave behind traditional mechanisms.

Once the proposed implementations have been done, an exhaustive comparison between the authentication mechanisms according to different criteria such as performance, usability or security level will be done.

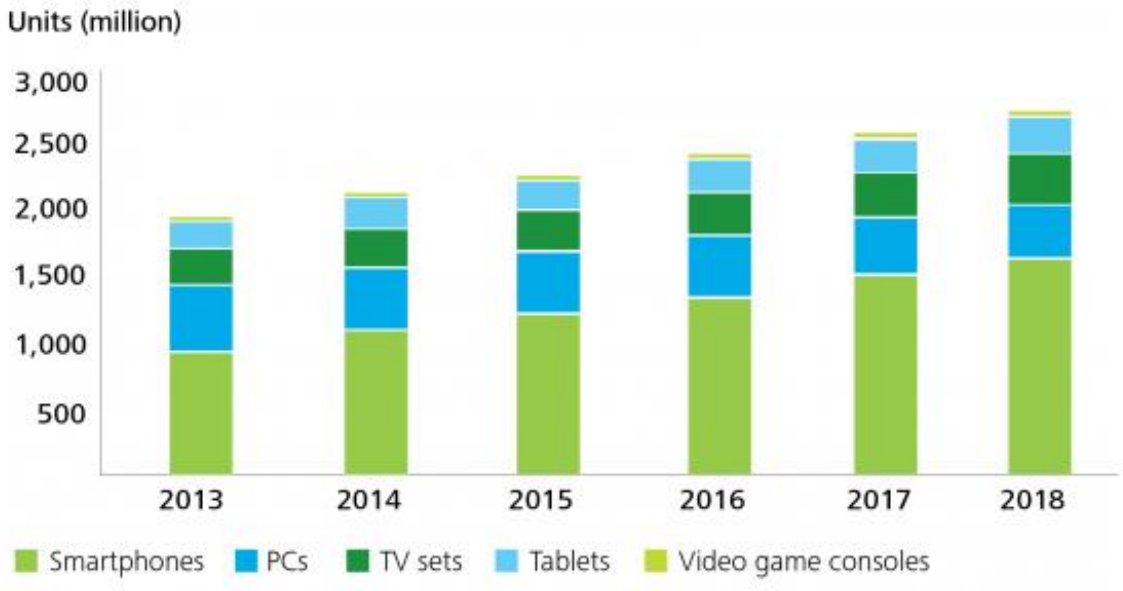
The overall objective of this paper is to make an implementation of several authentication mechanisms based on eyetracking technology, so it can be proved that there are authentication techniques with a reliable usability, safer than the actual ones. To achieve the main objective, the following **goals** will be stablished:

1. Make a **research** of the available authentication mechanisms that fit in this project. An analysis of those authentication mechanisms that could fit as a solution to the problem will be carried out.
2. Reach a depth **knowledge of the tracker** and the libraries that makes easier the connection between the device and the developer.
3. **Implement** the authentication mechanisms. In order to achieve a good usability, the code has to be clear and meaningful.
4. Once the application is done and works correctly, the application will be tested. Ten attemps of authentications are going to be made for each case and the **results** of every test will be recorded, the successful attemps, mean time used, security level….
5. Based on the results, a **comparison** between the authentication mechanisms will be done.

[1] To serve as an example, Apple bought recently an eye tracking company called SensoMotoric Instruments (SMI) and have hinted at the possibility of using this technology in future devices of the brand [1].

1.2 Socio-economic context

According to some studies, the prediction of the number of electronic devices that are going to be sold in the next year is going to keep increasing as every year [2]:



Ilustration 1: Combined global sales units of PCs, smartphones, tablets, TVs and video game consoles [2].

As we can see on the image, the most bought devices are the smartphones, which are also the devices that have the highest sales increase from one year to another. This increase is due to the fact that in this kind of electronic device, there is a larger range of prices. Nowadays, there are around 2.32 billion smartphone users and the number is expected to be 2.87 billion for the year 2020 [3].

It is easily seen that the incomes in the technological sector during the next years are going to be important. Therefore, more resources will be available for researchers in order to make advances in all of the technological environment, including authentication mechanisms. In addition to this, there will be a positive impact on society as companies will hire more qualified people for the technological development.

Technological progress is inevitable, as the use of devices has become a need to feel connected with the rest of society. The users are connected to the smartphones since they wake up in the morning until the end of the day. Knowing this, the number of companies dedicated to the technological world increases with the number of users, which lead to an economic impact in the different regions. An important part of the development of new technologies have as a goal to make our lifes easier and more secure.

The advantages are clear, but we have to talk about the disadvantages. No kind of dependency is beneficial for the members of society. Smartphone users are increasingly

younger because they refer to their parents, who use electronic devices constantly. The use of electronic devices can be dangerous because when you share multimedia content like photos or videos you will not know with total certainty who can get to view such content. Younger users may not even be aware of the danger this entails. Despite this, companies tend to manufacture products that fit the younger users encouraging their purchase and expanding their market.

The average user has approximately ninety accounts online which means that in the worst case, the user will have to remember ninety passwords. Owning so many accounts lead the users to use the same passwords on multiple accounts, although this compromise the security as one data breach can provide access to all of the accounts with the same password [4]. This can cause what is known as “password fatigue”, which can be described as a state of stress and exhaustion experienced by technology users who are overwhelmed by authenticationrelated demands on their time, energy and memory [5].

On average, password reset requests represents between 20% and 50% of all helpdesk calls, which leads to a decrease of the productivity as resetting passwords wastes a great amount of time and money. According to Forrester Research, making a single password reset costs around seventy dollars [6].

The development of the authentication mechanisms should tend to methods that are more natural or interfere minimally in the main task of the user, which is to use the service or device they want to access. Furthermore, it would not only have a healthy impact as users would be less stresed, but it would also have an economic impact.

1.3 Means employed

In relation to the resources that have been used during the different phases of the creation of this project, the tools that have been used and the role that each one of them has played are going to be detailed:

1.3.1 Hardware

To realize this project, it is going to be used the device made by The Eye Tribe company called The Eye Tribe EyeTracker.

One of the big advanteges of using this device, is the low cost that makes it a potentially interesting resource for research. In addition to this, the device provides features that ensure a great performance [7]:

- Sampling rate: 30 Hz to 75 Hz
- Accuracy: 0.5° – 1°
- Latency: < 1.6 ms

- Calibration: 6, 9, 12 points
- Working range: 45cm - 75cm
- Tracking area: 45cm x 30cm
- Screen sizes: Up to 24”
- SDKs in Java, C++ & C#
- Interface: USB3.0 type C

Ilustration 2: The Eye Tribe EyeTracker.

Unfortunately, the company reported on December 16, 2016 that they were going to stop developing eye trackers as they decided to go in a different direction with their technology [8].

However, it is still possible to acquire the device by other paths with the drawback of the official device support which no longer exists. The developer community will continue to be actively involved in creating projects and solving errors.

1.3.2 Software

The operating system that has been used is Microsoft Windows, and more specifically **Windows 10 Education**, which is the free version that university Carlos III provides to students.

It has also been used **Microsoft Word 2016** to write the memory. It is considered to be the most popular word processor in the world. It could be said that it is the most complete and powerful writing tool on the market and whenever it is necessary to work with digital documents, this is a very useful tool.

In addition to this, **Microsoft Visio 2013** has been used to make flow charts of the designed algorithms of the implemented authentication mechanisms.

1.3.2.1 Java

Java is a programming language and computing platform first released by Sun Microsystems in 1995. In this project, Java version 8 will be used to program and execute the applications made for each authentication mechanism.

Java Development Kit (JDK) version 8u121 has been used as the development environment for building applications and components that use Java programming language. The JDK includes useful tools for testing and fast fixing errors.

Java is the most used tool in the development of applications and programs, due to several factors:

- It is practical: it was designed to allow developers to get their job done with the minimum of fuss, enabling other developers to get their code at a later date and understand what it is supposed to do.
- Compatible with future versions: the written code for one version of Java will continue to run without any changes on newer versions.
- Java ensures scalability, performance and reliability.

1.3.2.2 The Eye Tribe SDK

The Software Development Kit version used for this project is the The Eye Tribe SDK 0.9.77. This not only includes a very complete API but also it is possible to use a software as an example, so that the developer can have it as a reference.

Once the SDK has been installed, two applications can be launched in order to know if everything is correct. The first one is called EyeTribe Server and it allows you to start the server. A connection with the device is tried and once this connection is made, it shows the traces of the state of the device. The second one is called EyeTribe UI and this is made to see the instructions of how to use it and after all, the user makes a calibration of the device.

This SDK is available not only for Java programming language but also for C ++ and C programming languages. Thanks to this SDK, the server can manage the coordinates of the screen that the user is staring at.

1.3.2.3 Eclipse

To implement the code of the Java aplications, it has been used Eclipse as the integrated development environment (IDE), which is the software where the Java code is going to be written and tested. This choice was made because Eclipse is written mostly in

Java and its primary use is for developing Java applications. As far as this project is concerned, the version Neon.3 will be used.

One of the main reasons for using Eclipse as the IDE, is the facilities that it provides to the user such as the file architeture of the project, the error assist while coding, code completion or the description of the available methods.

It is going to be necessary to download and install an extension into the Eclipse platform called JavaFX. JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy client applications [9].

1.3.2.4 JavaFX Scene Builder 2.0

As we claimed before, JavaFX will give the opportunity to the developer to work with graphics packages in order to design the different scenes of the application.

Thanks to JavaFX Scene Builder, it will be much easier to create and design the scenes used in the application. JavaFX Scene Builder is a visual environment where the developer can easily design user interfaces (UI) and the software generates itself the corresponding code into the FXML file where the scene is stored.

1.3.2.5 Libraries

For the project to go ahead, Eclipse provides some internal libraries called “JRE System Library”, where some interesting JavaFX libraries can be found.

In addition to this, one external library is going to be added to the project, called EyeTribeJavaFx, which includes several methods that are going to make the work easier. This library is easily found on the The Eye Tribe official site.

1.4 Memory structure

This document contains six chapters, where are going to include all the followed steps to reach the goal of the project, the implementation of several authentication mechanisms, creating them as Java applications, and then provide a comparison between these mechanisms using different parameters. A brief introduction of each chapter is provided down below:

- Chapter 1: Introduction. Motivation and goals of the thesis, the social and economic context, the software and harware used and the memory structure.

- Chapter 2: State of the art. A brief explanation of the authentication mechanisms that are going to be implemented.

- Chapter 3: General description of the authentication mechanisms implemented with the previous study and the design phase.

- Chapter 4: Results and comparison of the created mechanisms.

- Chapter 5: Thesis planning and budgets. Description and dates of the different phases of development of this thesis and the cost of the resources.

- Chapter 6: Conclusion. Analysis and conclusions obtained from this project. Future developments of the topic are included as well.

Capitulo 2

Estado del arte

2.1 Tecnologías utilizadas

2.1.1 Tecnología Eyetracking

Se puede definir ***eyetracking*** como aquella tecnología que tiene como fin extraer información del usuario teniendo en cuenta los movimientos oculares que este realiza. Para dar vida a esta tecnología, las empresas del sector fabrican unos dispositivos que contienen sensores y algoritmos. El dispositivo genera un patrón de luz infrarroja y gracias a los sensores incorporados en él, se consiguen capturar los fotogramas de los ojos del usuario y el patrón de reflexión para posteriormente procesar los datos recogidos y finalmente, mediante un algoritmo matemático, se calcula la posición de los ojos y la mirada del usuario frente a la pantalla [10]. Estos dispositivos reciben el nombre de *eyetrackers*.

2.1.1.1 Composición de un *eyetracker*

Un dispositivo *eyetracker* está compuesto de los siguientes elementos [11]:

- **Micro proyectores avanzados** que se utilizan para crear un patrón de reflexión de rayos infrarrojos en los ojos.
- **Sensores** encargados de captar imágenes de alta tasa de fotogramas de los ojos del usuario y también los patrones de reflexión.
- **Algoritmos de procesamiento de la imagen**. El sistema es inteligente ya que es capaz de encontrar detalles específicos en los ojos de los usuarios y en los patrones de reflexión, permitiendo interpretar la imagen generada por los sensores. Estos algoritmos calculan la posición de los ojos y de la mirada del usuario en la pantalla.
- **Aplicaciones orientadas al usuario**. Una aplicación inteligente es añadida para habilitar las distintas formas de utilizar esta tecnología.

2.1.1.2 Aplicabilidad *eyetracking*

La tecnología *eyetracking* presenta numerosas aplicaciones que se pueden utilizar para una gran variedad de áreas de estudio, desde marketing hasta empresas deportivas. Algunas de las aplicaciones más relevantes son:

- **Publicidad y patrocinio**. Gracias a la tecnología *eyetracking*, se puede evaluar la visibilidad y el atractivo general de cualquier anuncio, determinando si los anuncios captan la atención de los lectores [12].
- **Diseñadores web**. De igual forma que para la publicidad y los patrocinadores, a los desarrolladores web les interesa saber que puntos de una página son más visionados por los usuarios para colocar en ellos elementos deseados.
- **Rendimiento deportivo**. Es posible utilizar la tecnología *eyetracking* para optimizar características asociadas a deportistas. Como ejemplo, si un atleta quiere entrenar sus reflejos y su tiempo de reacción, se puede implementar un mecanismo que genere un punto móvil y que pueda entrenar y evaluar al atleta [13].
- **Accesibilidad para personas discapacitadas**. Una persona discapacitada que no tenga movilidad en los brazos tiene numerosas limitaciones, entre las que se encuentra el uso del ordenador. Con esta tecnología se puede llegar a conseguir que estas personas sean capaces de navegar por Internet e incluso jugar a videojuegos gracias a la mirada [13].
- **Investigación psicológica**. En esta área de estudio, la atención visual puede asociarse a la forma en la que un cerebro funciona. Con la tecnología *eyetracking* se puede hacer un estudio para la asociación de patrones visuales con cualquier tipo de desorden de salud mental [14].
- **Autenticación**. Esta tecnología puede utilizarse como sistema de autenticación de usuarios. Aporta un aumento en la seguridad del proceso de autenticación y además es amigable de cara al uso del usuario [15].

2.1.1.3 Comparación de dispositivos en el mercado

En la actualidad existe en el mercado una gran competencia relacionada con la tecnología *eyetracking*. Es un área que sigue siendo joven y que está en continua evolución. A continuación, se van a mostrar algunos de los dispositivos más utilizados y se realizará una comparación de sus características más significativas.

DISPOSITIVO	OS	PRECIO	HW	VELOCIDAD [Hz]	LENGUAJES	TAMAÑO PANTALLA
The Eye Tribe [7]	Android, Windows, OSX	90 €	Dispositivo	30 - 75	C++, C#, Java	Hasta 24”
Tobii Pro X3-120 [16]	Windows 7/8.1/10	150 €	Dispositivo	120	C++, .Net	Hasta 25”
AEye eye tracker [17]	Windows, Android	180 €	Dispositivo	40 - 200+	C++, C#	Hasta 24”
SentiGaze [18]	Windows 2003/XP/Vista/7/8	890 €	Webcam	30	C++, C#, Visual Basic	Hasta 24”
GP3 *Eye Tracker* [19]	Windows 7/8.1/10	695 $	Dispositivo	60	Cualquiera que soporte comunicaciones TCP/IP	Hasta 24”
Pupil [20]	Linux, MacOS, Windows	1.640 €	Lentes	120	Phyton, C, C++	Hasta 24”

Tabla 1: Comparación dispositivos en el mercado

Como se puede apreciar, la opción que ofrece unas prestaciones aceptables a un menor precio es el dispositivo *The Eye Tribe*, lo que convierte este dispositivo en el ideal para realizar trabajos de investigación a bajo coste.

2.1.2 Java

2.1.2.1 JDK y JRE

La implementación de los mecanismos de autenticación se va a realizar mediante aplicaciones diseñadas con el lenguaje de programación orientada a objetos Java. Para conseguir que este lenguaje sea entendible por el equipo en el que se va a desarrollar el código, es necesario tener instaladas unas herramientas. El *software* imprescindible para un desarrollador es ***Java Development Kit*** (JDK). Este *software* proporciona herramientas de desarrollo para la creación de aplicaciones en Java. Las herramientas más relevantes para un desarrollador son [21]:

- **Javac.** El compilador de java para traducir el código fuente a una serie de instrucciones a nivel interno.
- **Jdb**. La herramienta de debug con la que el desarrollador puede probar paso a paso la ejecución de la aplicación, pudiendo observar la variación de los valores de las variables y detectar errores de manera óptima.
- **Java.** Sirve para cargar las aplicaciones Java. Se encarga de interpretar los ficheros que contienen las clases generadas por el compilador.

El *software* JDK incluye una herramienta que se encarga de la ejecución de cualquier aplicación Java, llamada **Java Runtime Environment** (JRE). Este *software*

contiene todos los paquetes necesarios para que un programa Java pueda ejecutarse. Algunos de los componentes del *software* JRE son [22]:

- Tecnologías de implementación.
- Herramientas de interfaz de usuario
- Integración de librerías
- Máquina virtual de Java

A pesar de que el JDK incluye el JRE, un usuario que solo quiera el *software* JRE se lo puede descargar e instalar individualmente.

2.1.2.2 JavaFX

El presente Trabajo de Fin de Grado está basado en una extensión de Java llamada JavaFX. Este *software* es un conjunto de paquetes de contenido gráfico y de medios que al descargarlos e instalarlos en el entorno de trabajo, ofrece al desarrollador la posibilidad de diseñar, crear, probar e implementar el contenido de una aplicación de una manera más cómoda.

Los beneficios de utilizar JavaFX para el desarrollo de una aplicación Java son numerosos ya que tiene una gran variedad de herramientas que complementan a JavaFX y que son de gran utilidad. A continuación, se van a detallar algunas de dichas herramientas utilizadas en JavaFX [23]:

- Un nuevo formato llamado **FXML**, el cual se utiliza exclusivamente para definir como va a ser la interfaz visual.
- Existe una aplicación de gran utilidad llamada **Scene Builder** que se puede integrar dentro de Eclipse y que facilita la creación y la modificación de los documentos FXML de manera que el desarrollador pueda mantener el control de la escena que está creando visualmente sin necesidad de escribir código.
- Se puede utilizar **código CSS** (Cascading Stylesheets) en una aplicación JavaFX para controlar y dar el estilo deseado al contenido de la aplicación.
- Proporciona una **librería integrada** para gráficas, tanto en 2D como en 3D, y para herramientas de animación.
- Nuevas **funcionalidades** para la edición de interfaces, más controles y un enriquecimiento general de la interfaz del usuario. Aumenta así la capacidad de los desarrolladores para innovar en futuras aplicaciones.

2.2 Técnicas de autenticación

En los últimos años se han propuesto las contraseñas gráficas como mejora a la seguridad en la autenticación de los usuarios en los dispositivos. El nivel de seguridad del proceso de autenticación aumenta considerablemente gracias a la tecnología *eyetracking*, y este es el motivo por el que se ha decidido trabajar con esta tecnología en el presente Trabajo de Fin de Grado.

Con el uso de esta tecnología, no solamente se consigue una mayor usabilidad, ya que se suprime el uso del teclado o la mano para realizar la acción, sino que además disminuye el riesgo de que se produzcan ciertos tipos de ataques.

A continuación, se van a detallar los diferentes sistemas de autenticación que se pueden implementar con la tecnología *eyetracking*.

2.2.1 Mecanismos de autenticación basados en el movimiento

En este grupo se incluyen las técnicas de autenticación más utilizadas que son la contraseña **alfanumérica** y el **código PIN**. A pesar de ser las más utilizadas, desde sus inicios han estado expuestas a posibles ataques, como puede ser el intento de suplantación o el robo de la contraseña. El uso tradicional de estas técnicas ha reportado numerosos casos de usuarios que han tenido dificultades para recordar las contraseñas ya que puede ser que no se utilicen durante un largo periodo de tiempo. Recientemente, varios estudios han demostrado que las contraseñas gráficas son más sencillas de recordar ya que los humanos tienen esa habilidad proficiente. Es por eso que existe una propuesta para implementar estos mecanismos de autenticación mediante la tecnología *eyetracking* [15].

A la hora de realizar la autenticación de los mecanismos alfanuméricos mediante la tecnología *eyetracking*, el funcionamiento es similar y lo que varía de cara al usuario es que la contraseña se introduce mirando al teclado virtual situado en la pantalla en lugar de escribirla manualmente. Los *layouts* de los teclados diseñados para este proyecto son parecidos a los que utilizan los *smartphones*, contienen todos los caracteres del abecedario español, menos la “ñ”, y los diez digitos. El teclado virtual se introduce con formato imagen dentro de las escenas necesarias y una vez incrustrada en la escena se desarrolla el algoritmo. Para este tipo de autenticaciones es recomendable que el diseño de los teclados esté hecho pensando en la memoria visual de los usuarios de manera que no tengan problemas para encontrar los caracteres en el teclado.

Cuando el usuario esté creando la contraseña o cuando esté probando la autenticación, tendrá un teclado virtual en la pantalla y tendrá que ir focalizando la mirada en los caracteres que formen la contraseña. Con la ayuda del *eyetracker*, la aplicación tiene la posibilidad de gestionar los rangos de coordenadas definidos para cada carácter. Entre

carácter y carácter es recomendable que la aplicación proporcione un margen temporal para facilitar la usabilidad. Una vez que el usuario finaliza la creación de la contraseña, la aplicación habrá almacenado la contraseña en el sistema mediante la concatenación de los caracteres, para compararla con la contraseña que introducirá el usuario cuando intente autenticarse.

Se ha encontrado un estudio sobre el mecanismo alfanumérico aplicado a la escritura virtual de varios textos, proceso que realizaron varias personas, en el que se reflejó que la media individual de error a la hora de seleccionar un carácter fue del 0,83% [24]. Este porcentaje confirma que la implementación de mecanismos de autenticación de tipo tradicional mediante *eyetracking* es viable, ya que es una tasa de error baja.

Otro de los mecanismos implementables con esta tecnología es el **patrón de movimiento**. Esta técnica es muy utilizada en el desbloqueo de dispositivos móviles y para darle un uso más seguro se puede desarrollar junto a la tecnología *eyetracking*. No obstante, el nivel de seguridad que puede ofrecer sigue siendo inferior a otros mecanismos de autenticación.

Para este caso, se presenta una interfaz que contiene una serie de bloques separados por líneas visibles. El usuario deberá visualizar varios bloques de manera continuada, formando una trayectoria como contraseña. El dispositivo recogerá los valores de la mirada y la aplicación los transformará en un número que representará al bloque del punto visualizado. Una vez establecida la contraseña, el sistema habrá almacenado los identificadores de los bloques que forman la contraseña para compararlos posteriormente con los bloques seleccionados por el usuario al intente autenticarse.

2.2.2 Mecanismos de autenticación basados en el movimiento del ojo frente a imágenes

Estas técnicas de autenticación requieren el uso de una o varias imágenes que se presentan al usuario al inicio de una sesión.

Se ha demostrado la posibilidad de implementar mediante tecnología *eyetracking* el mecanismo de autenticación denominado "**Passfaces**". En esta técnica, la interfaz está compuesta de varias imágenes de rostros humanos, normalmente en forma de matriz 3x3. El usuario debe visualizar las caras seleccionadas en fases anteriores para la autenticación. Esta técnica se basa en la suposición de que las personas pueden recordar caras humanas mejor que otras imágenes. No obstante, se puede implementar esta misma técnica mediante *eyetracking* con imágenes de cualquier tipo. Estas técnicas ofrecen la posibilidad de aumentar el nivel de seguridad al realizar el proceso de autenticación dos veces alterando la posición de las imágenes en ambos casos [15].

Este mecanismo presenta al usuario una interfaz con nueve imágenes de rostros humanos con características distintas, como puede ser un rostro sonriente, otro serio, otro llorando, etc. Cada imagen tiene asociado un valor que la identifica y el usuario deberá visualizar cuatro de las nueve imágenes como contraseña, quedando registrado en el sistema los identificadores de dichas imagenes. A la hora de autenticarse, el usuario debe realizar el proceso de identificación de las imágenes que forman la contraseña dos veces, ya que en ambos casos la posición de todas las imágenes en la interfaz es distinta.

Otra técnica de autenticación consiste en utilizar una **imagen** de la naturaleza, seleccionada por el usuario, en la que la contraseña estará formada por varios puntos de la imagen. Los rasgos u objetos visuales que puedan servir de referencia dentro de la imagen facilitan el uso de esta técnica [20].

La aplicación que implementa este mecanismo, va a solicitar al usuario que visualice un número determinado de puntos en la imagen que se haya escogido. Cada punto visualizado que forme parte de la contraseña va a estar representado por sus coordenadas de la pantalla, y serán estas las que se almacenen en el sistema. Cuando se hayan visualizado todos los puntos necesarios, el sistema tendrá almacenadas, de manera organizada, todas las coordenadas para posteriormente ejercer una comparación de coordenadas cuando el usuario intente autenticarse. Para mejorar la usabilidad es recomendable el uso de un margen de error a la hora de realizar la comparación de coordenadas.

En 2004 se llevó a cabo una investigación sobre el mecanismo de autenticación basado en la visualización de varios puntos dentro de una imagen. En dicho estudio se estableció un umbral de tiempo de fijación para que, una vez fuese alcanzado ese tiempo a la hora de visualizar los puntos que forman la contraseña, se guardara la posición. Los resultados del estudio mostraron que el rango de tiempos de fijación funcionales era entre 0,1 y 1,3 segundos, debido a que en ese rango la tasa de reconocimiento era cercana al 100% [25].

2.2.3 Mecanismos de autenticación basados en la lectura de textos

Existen estudios sobre la posibilidad de utilizar como mecanismos de autenticación la lectura de un texto. Para distinguir a los usuarios se utilizan parámetros como la rapidez de lectura del texto, en qué palabras emplea más tiempo, si repite la lectura de algunas palabras, etc. Se ha demostrado que es una técnica bastante precisa ya que es difícil suplantar la forma de lectura de un sujeto [26].

El funcionamiento de este mecanismo es sencillo, se va a mostrar por pantalla un texto que el usuario debe leer de la manera más natural posible. El dispositivo recogerá los datos necesarios para dictaminar si la autenticación ha sido realizada por el dueño del

dispositivo o no. En cuanto a la precisión que ofrece el mecanismo, según los resultados de un estudio hay un 0,17% de tasa de falsa aceptación, un 7,89% de tasa de falso rechazo y un 0,58% de tasa de error medio [26].

El gran inconveniente que presentan estas técnicas de autenticación es la baja usabilidad ya que el usuario deberá leer el texto siempre que quiera iniciar sesión. En consecuencia, se puede intentar que la usabilidad mejore proponiendo textos cortos como contraseña, pero cuanto más breve sea el texto, más baja será la seguridad.

En la actualidad hay aplicaciones que utilizan estas técnicas de autenticación como, por ejemplo, la que diseñó Keith Rayner y que explica en su artículo "Eye movements in reading and information processing" [26].

2.2.4 Selección de las técnicas de autenticación

Las técnicas de autenticación basadas en la tecnología *eyetracking* mencionadas anteriormente, ofrecen un nivel de seguridad y de usabilidad mayor en comparación con las técnicas tradicionales.

La idea inicial para la realización de este proyecto es escoger aquellas técnicas que, a priori, tengan características distintas para poder establecer una comparación entre los resultados obtenidos con el uso del dispositivo *The Eye Tribe EyeTracker*.

Se han decidido implementar los siguientes mecanismos de autenticación: **Patrón de movimiento, código PIN, código alfanumérico** e **imagen fija**.

2.3 Restricciones y marco regulador

En este apartado se va a analizar el marco legal que afecta a este Trabajo de Fin de Grado. Este proyecto no utiliza ningún dato personal que pudiera suponer implicaciones pertenecientes a la Ley Orgánica de Protección de Datos (LOPD) [27].

En cuanto a las licencias que se han utilizado durante el desarrollo de la aplicación, únicamente se ha utilizado la proporcionada por The Eye Tribe que permite el uso de su *software*. Esta licencia solamente tiene limitaciones comerciales, lo que significa que las aplicaciones desarrolladas con este *software* no pueden ser comercializadas.

En referencia a las restricciones que surgen en este Trabajo de Fin de Grado, es necesario tener el *hardware* y el *software* de The Eye Tribe, cuyo valor es de 90 € y que incluye la licencia de uso gratuita ya que no se ha utilizado para fines comerciales [28]. También es necesario cumplir los requerimientos del sistema, entre los que se encuentra

tener instalada la versión de Java 7 o superior. Para ello es necesario aceptar el acuerdo de licencia de Oracle: “Oracle Binary Code License Agreement for Java SE and JavaFX Technologies” [29].

Capítulo 3

Implementación

3.1 Arquitectura genérica

En este capítulo se va a explicar cómo se han realizado las implementaciones de los mecanismos de autenticación que han permitido la creación de la aplicación basada en la tecnología de seguimiento de ojos conocida como *eyetracking*.

Se ha llevado a cabo una investigación con el fin de averiguar qué mecanismos de autenticación implementados con tecnología *eyetracking* han sido diseñados e implementados, de tal forma que usaremos los más interesantes para que formen parte de este proyecto. Los mecanismos de autenticación elegidos para ser implementados son: patrón de movimiento, código numérico, código alfanumérico e imagen fija

La estructura que tiene la aplicación a nivel general se refleja en la Ilustración 3:

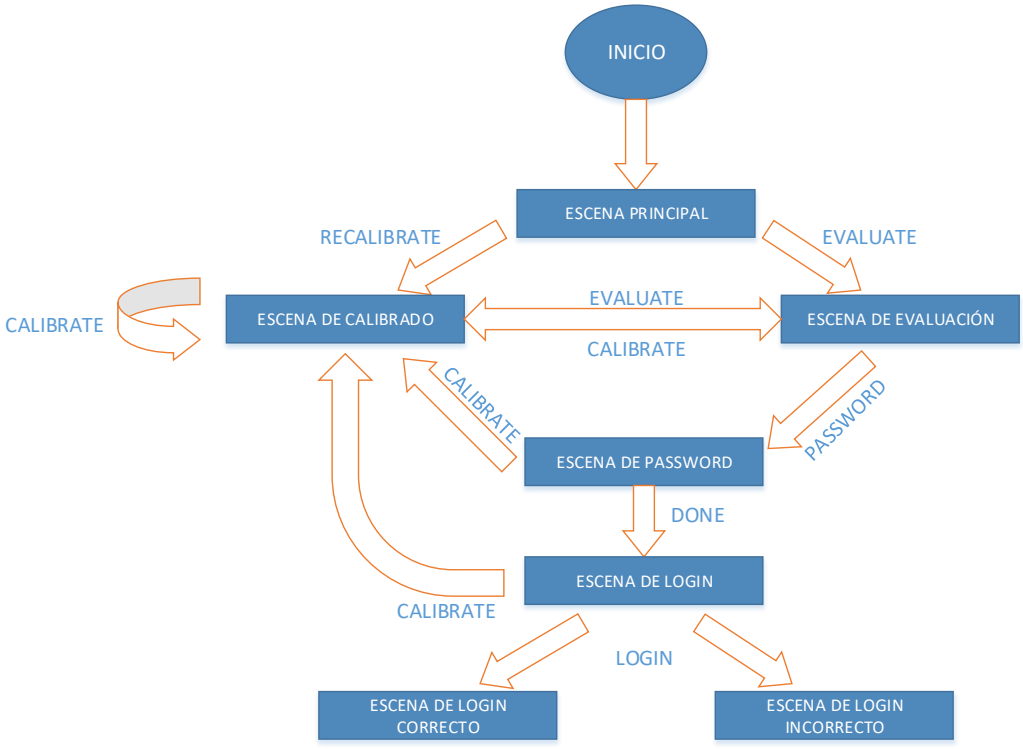


Ilustración 3: Arquitectura de la aplicación.

Cada uno de los bloques azules de la Ilustración 3 se corresponde a una escena, la cual representa un menú dentro de la aplicación y cada una de ellas está recogida en un fichero con extensión FXML, que es el formato de las escenas en JavaFX. Las escenas de *password* y de *login* son las más importantes en cuanto a la implementación de la autenticación, ya que en ellas se ejecutan los algoritmos diseñados para la autenticación en sí. Cada una de las escenas va a tener asociada una clase Java en la que se implementará la funcionalidad de dicha escena. A continuación, se va a mostrar la tabla donde se puede ver la relación de nombres entre las escenas y las clases y posteriormente se detallará cada escena:

Fichero de la escena (.fxml)	Clase de la escena (.java)
scene_main	SceneMainController
scene_calibrate	SceneCalibrationController
scene_evaluate	SceneEvaluationController
scene_password	ScenePassword
scene_login	SceneLogin
scene_login_result	SceneLoginResult
scene_login_result2	SceneLoginResult2

Tabla 1: Relación de nombres entre escenas y clases.

- **Escena principal:** Al iniciar la aplicación, este el primer menú con el que se encuentra el usuario. Esta escena contiene una animación en el centro de la pantalla que muestra la posición de los ojos del usuario respecto al dispositivo. Además, para comprobar desde un inicio si el dispositivo está bien calibrado, hay un circulo blanco que se mueve por la pantalla en función del punto que el usuario esté mirando en ese instante. También tiene dos botones, `recalibrate` y `evaluate`, situados en las esquinas inferiores del menú mediante los cuales el usuario podrá dirigirse a la escena de calibrado o a la escena de evaluación.

Ilustración 4: Escena principal.

- **Escena de calibrado:** En caso de que el usuario perciba que el dispositivo no está reconociendo la mirada de forma adecuada, podrá acceder a esta escena desde cualquier punto de la aplicación. Cuando el usuario entra a este menú, se muestra un texto indicando que se debe seguir un círculo (“Follow the circle”). A continuación, aparecerá un círculo por la pantalla que irá cambiando de posición cada breve periodo de tiempo. Una vez que el círculo haya hecho el recorrido completo, se hace una valoración de la calibración en forma de estrellas en la esquina superior derecha, mostrándose 4 si ha sido completamente satisfactoria y 0 si ha sido insuficiente.

Ilustración 5: Escena de calibrado.

- **Escena de evaluación:** Este menú resulta de gran utilidad para el usuario, ya que en él podrá evaluar la precisión del dispositivo en ese preciso instante. Aparecerán nueve círculos repartidos por la pantalla, en forma de matriz tres por tres, de manera que cuando el usuario visualice alguno de ellos el círculo que originalmente es de color blanco pasará a tener un color rojo. El color rojo será más intenso cuando el usuario fije la mirada en el centro exacto del círculo. Como vemos en la imagen inferior, esta escena tiene dos botones, `calibrate` para redirigir al usuario a la escena de calibrado y `password` para redirigir al usuario a la escena de creación de la contraseña.

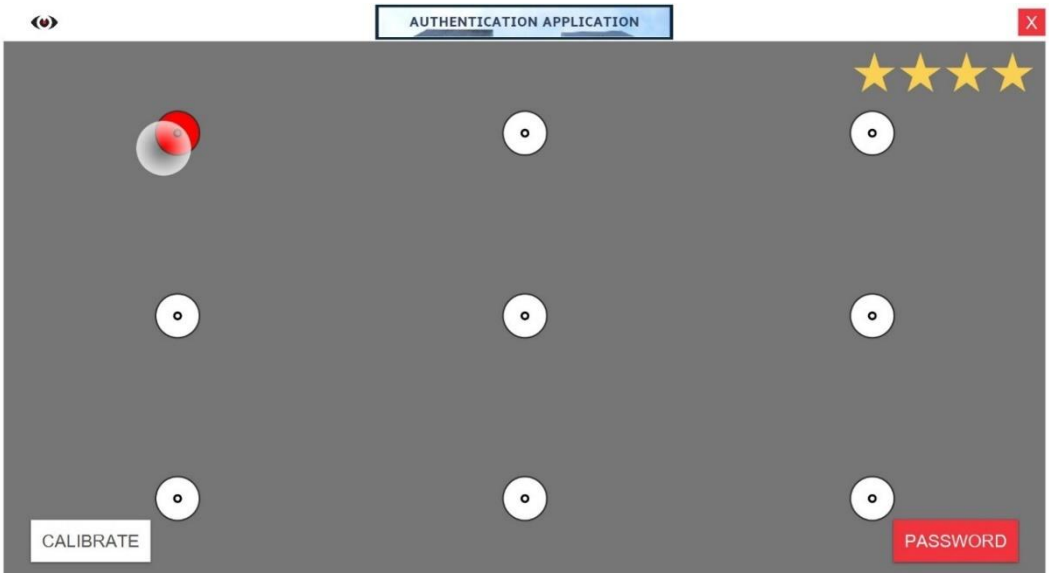


Ilustración 6: Escena de evaluación.

- **Escena de *password*:** En función del mecanismo que se esté empleando, el usuario deberá establecer la contraseña oportuna. Una vez se haya generado la contraseña, el botón `login` situado en la esquina inferior derecha redirigirá al usuario a la escena de *login*. A continuación, se muestra el diagrama de flujo correspondiente al algoritmo diseñado para el establecimiento de la contraseña:

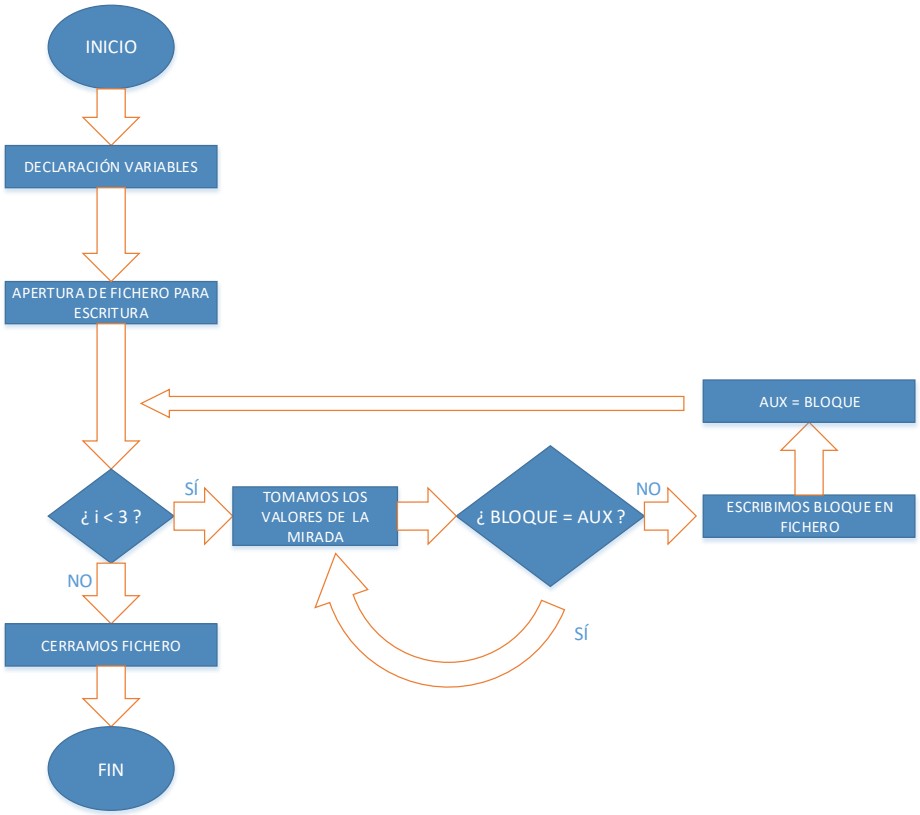


Ilustración 7: Diagrama de flujo de proceso de creación de la contraseña.

- **Escena de *login*:** Esta escena será la última en la que el usuario deberá interactuar con el dispositivo. Se deberá introducir la contraseña establecida en la escena de *password*. Una vez introducida la contraseña el usuario deberá presionar el botón situado en la esquina inferior derecha denominado `done`. A continuación, se muestra el algoritmo diseñado para realizar la autenticación:

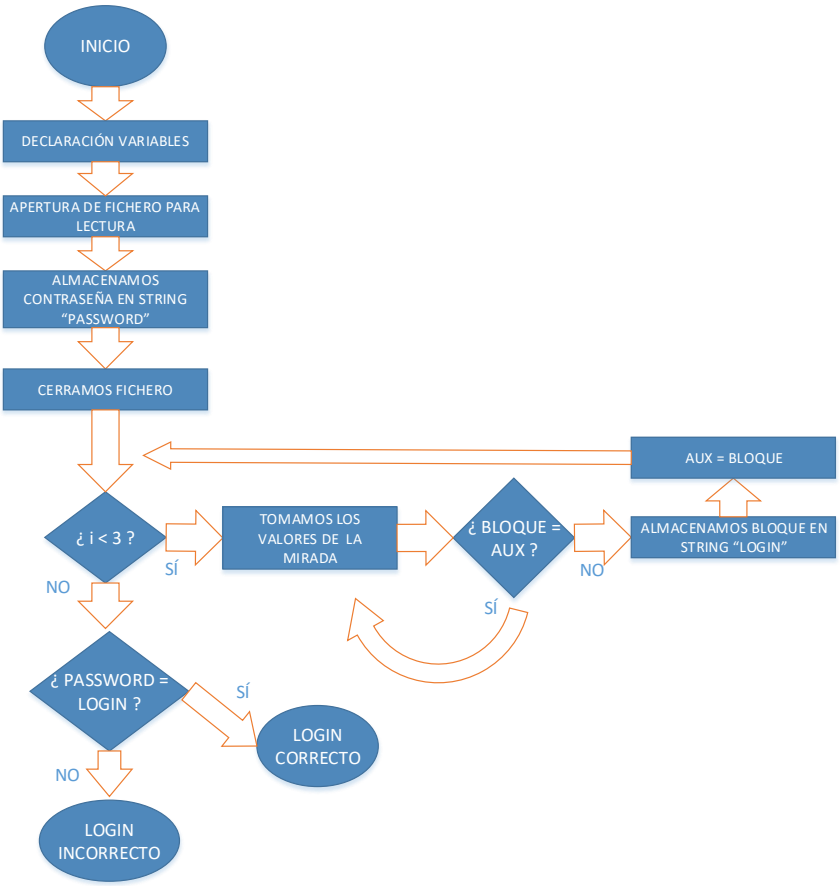


Ilustración 8: Diagrama de flujo de proceso de login.

- **Escena de *login* correcto o incorrecto:** Al presionar el botón `done` en la escena de *login*, la aplicación redirigirá al usuario automáticamente a la escena de *login* correcto si la autenticación ha sido correcta o a la escena de *login* incorrecto si la autenticación no ha sido satisfactoria. Además, estas escenas tienen unos bordes de color distinto ya que en ellas se acaba la aplicación.

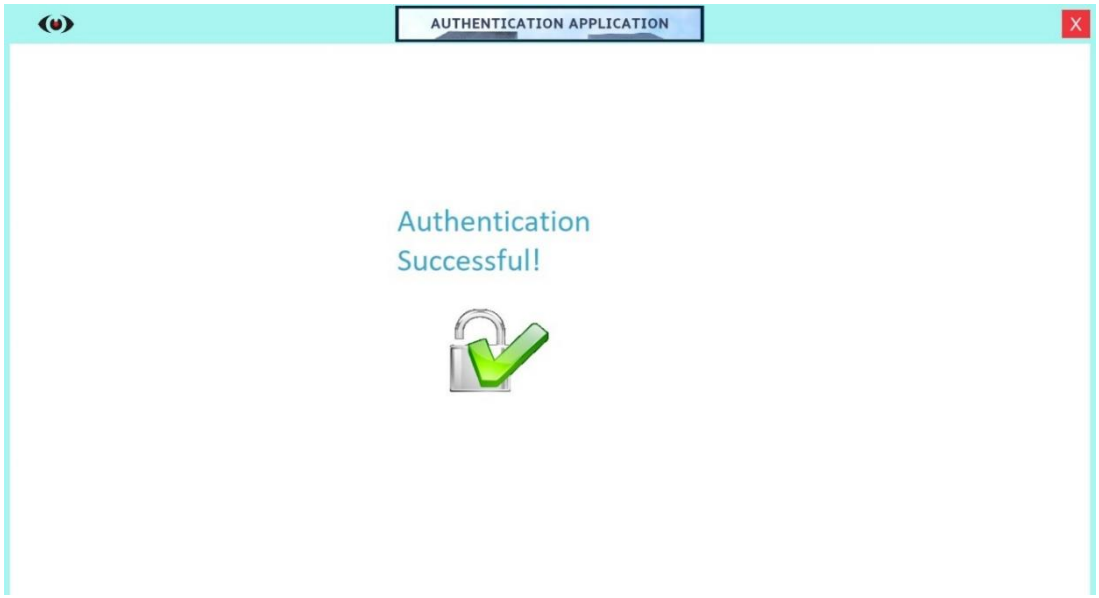


Ilustración 9: Escena de login correcto.

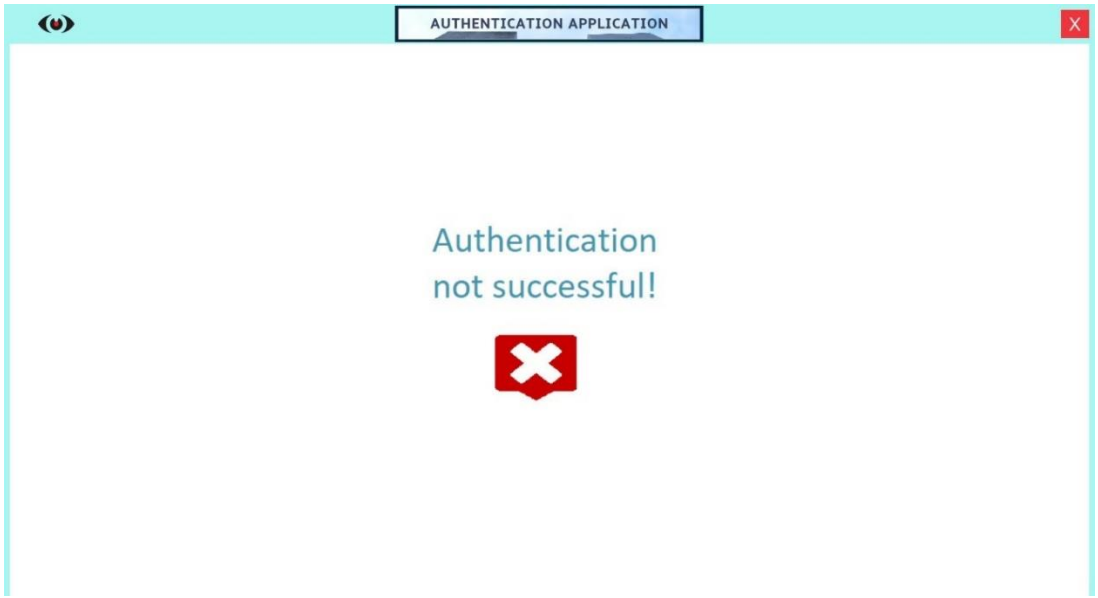


Ilustración 10: Escena de login incorrecto.

En relación a las **características técnicas** de las aplicaciones diseñadas, la estructura de ficheros que tienen todos los proyectos es igual. Los ficheros están divididos en cuatro paquetes:

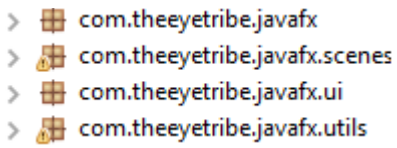


Ilustración 11: Estructura de ficheros de la aplicación.

En el primer paquete, `com.theeyetribe.javafx`, están contenidos los ficheros necesarios para el arranque de la aplicación. Destaca la clase `Main.java` donde está el código que gestiona las tareas que la aplicación debe realizar y también están las imágenes incluidas dentro de la interfaz de la escena inicial.

En el segundo paquete, `com.theeyetribe.javafx.scenes`, es donde se sitúan los ficheros que contienen el código de cada una de las escenas de la aplicación. Cada escena tendrá sus funcionalidades implementadas en estos ficheros Java. Cabe destacar la clase `SceneController.java`, ya que esta clase contiene varios métodos que comparten todas las clases incluidas en este paquete.

El paquete `com.theeyetribe.javafx.ui` está compuesto por dos ficheros Java que tienen determinadas funcionalidades dentro de la aplicación. El primer fichero se llama `CalibrationButton.java` y contiene el código necesario para que aparezca dentro de las escenas el circulo blanco que simboliza el punto que el usuario está mirando. El segundo fichero se llama `GazePane.java` y su funcionalidad es la de modificar el color del punto de referencia de la escena de evaluación, el color pasa de blanco a rojo cuando el usuario visualiza dichos puntos.

El último paquete de ficheros, `com.theeyetribe.javafx.utils`, contiene clases Java que se encargan de gestionar las comunicaciones entre el dispositivo y el equipo y también de verificar el estado de los parámetros: calcular la velocidad de conexión, los *frames* por segundo, las coordenadas de los puntos visualizados, la calidad del calibrado actual, gestionar los puertos del dispositivo…

3.2 Patrón de movimiento

Para comenzar a implementar este mecanismo, se ha tenido que diseñar la escena de *password* y *login* con las que el usuario va a interactuar. Para ello se ha realizado una división por bloques separados por líneas visibles sobre el fondo de la aplicación, quedando como una matriz 3x4. Además, el usuario podrá visualizar el círculo blanco que representa el punto exacto en el que esté focalizando su mirada.

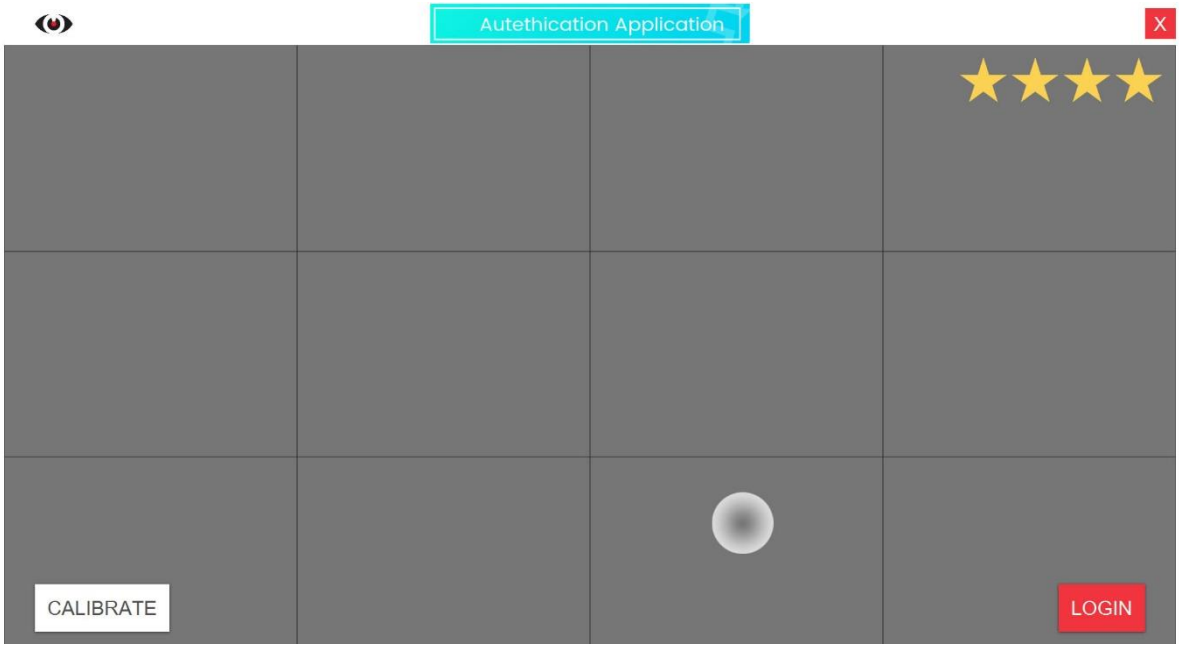


Ilustración 12: Escena password/login del patrón de movimiento.

Para realizar la división por bloques, se han utilizado los atributos anchura y altura de la escena, ambos proporcionados por la API de JavaFX.

Una vez diseñada la escena, hemos desarrollado un método en las clases `ScenePassword` y `SceneLogin` para poder controlar el punto exacto al que el usuario está mirando. El nombre del método es `tablero` y va a devolver un valor de tipo Integer, el cual se corresponde al identificador del bloque que el usuario está observando. A este método se le proporciona por parámetros las coordenadas del punto que el usuario está mirando y él se encargará de transformar las coordenadas a un valor identificador del bloque.

El método que va a iniciar la ejecución del algoritmo dentro de la aplicación se llama `initGazeImages`. Cuando el hilo de la aplicación acceda a este método, se va a generar un fichero, el cual se configurará con permisos de escritura o lectura en función de si el usuario está generando la contraseña dentro de la escena `ScenePassword` (escritura en el fichero) o si por el contrario está realizando la autenticación en la escena `SceneLogin` (lectura del fichero). Este fichero funciona a nivel interno y es completamente ajeno al usuario.

En este punto el usuario estará situado en la escena `ScenePassword` y deberá interactuar con el dispositivo, ya que tendrá que observar de manera consecutiva tantos bloques como se hayan definido necesarios para la contraseña.

Una vez que la aplicación haya capturado los valores de las coordenadas del punto que el usuario está visualizando, se ejecutará el método `tablero` para obtener el

identificador del bloque en el que se encuentra. Además, se escribirá en el fichero el valor del identificador. Este proceso se repetirá hasta que se hayan obtenido todos los identificadores de los bloques que componen la contraseña.

Terminada la creación de la contraseña, el usuario accederá a la escena `SceneLogin` dónde realizará la autenticación. En este momento la aplicación está diseñada para volcar la contraseña que contiene el fichero a una variable de tipo String denominada `password`.

El usuario deberá visualizar los bloques en el mismo orden en el que los observó durante la creación de la contraseña para que la autenticación sea correcta. En este caso, los identificadores de los bloques visualizados en la escena `SceneLogin` se almacenarán en una variable de tipo String llamada `loginTemporal`. Una vez que el usuario haya terminado de realizar la autenticación, se va a hacer una comparación entre los Strings `password` y `loginTemporal`. Si son iguales, la autenticación habrá sido satisfactoria y si son distintos la autenticación habrá fracasado.

Otras observaciones:

- Los bloques de la contraseña deberán ser contiguos.
- Para escribir en el fichero un valor de tipo Integer con forma de String, se ha utilizado `String.valueOf(bloque)`.
- Para comparar las dos variables de tipo String se ha utilizado la función `equalsIgnoreCase()`.

3.3 Código numérico

Uno de los sistemas más utilizados para identificar y autenticar a una persona en cualquier dispositivo es el código PIN. No obstante, la forma en que se utiliza la autenticación mediante el código PIN presenta numerosas vulnerabilidades. Gracias a la implementación de este sistema mediante tecnología de reconocimiento de la mirada, el número de posibles ataques se reduce considerablemente.

Para empezar a implementar este mecanismo de autenticación, se ha diseñado la escena de contraseña y *login*:

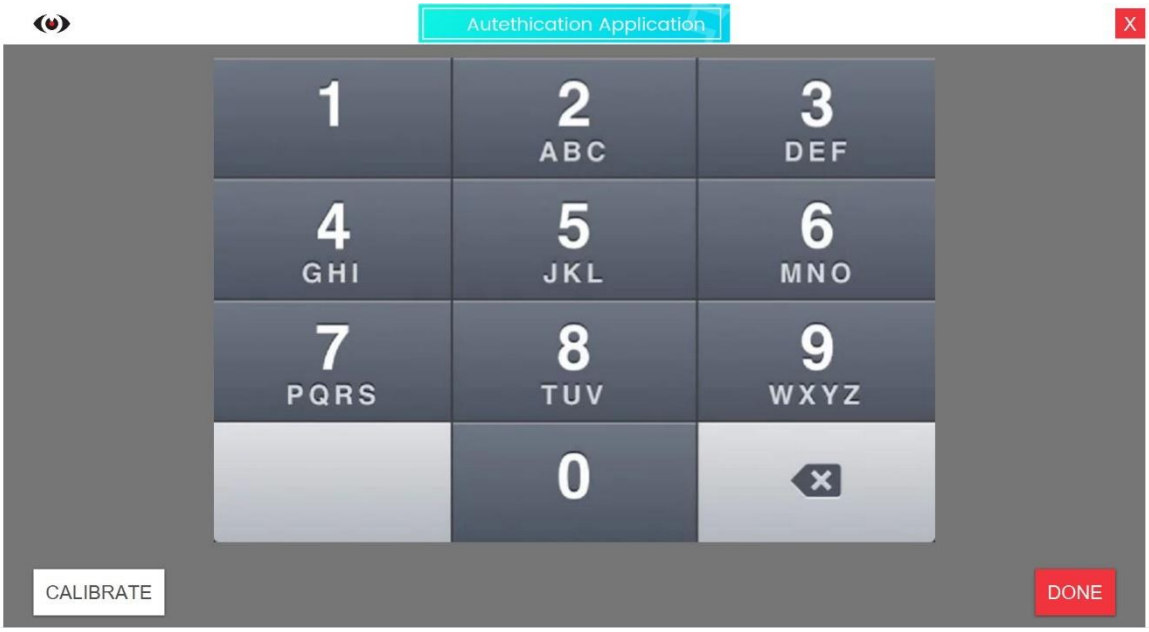


Ilustración 13: Escena password/login de código PIN.

En esta ocasión se ha averiguado cuales son las coordenadas de los píxeles que limitan unos números con otros. Para ello nos hemos ayudado de un sencillo programa externo, llamado **Mofiki's Coordinate Finder**, el cual nos ha facilitado esta labor. Las regiones con las que se van a trabajar en este sistema son:

Ilustración 14: Coordenadas de código PIN.

Una vez que se ha diseñado la escena con la que el usuario va a interactuar y se han averiguado las coordenadas que van a delimitar las regiones de los números, se ha decidido reutilizar el método `tablero`, diseñado para el mecanismo de autenticación del patrón de

movimiento. Para este caso, cada número observable de la escena va a tener asociado un identificador con el mismo valor.

Cuando se inicia la aplicación y el usuario llega a la escena de contraseña, se inicializa el método `initGazeImages` donde está el código referente al algoritmo diseñado para la autenticación.

En cuanto a la gestión de las coordenadas visualizadas, se ha utilizado una variable de tipo Point2D que facilita la API de JavaFX. Representa un punto geométrico de coordenadas de dos dimensiones. En esta variable se va a recoger el valor de la mirada del usuario y mediante un método que proporciona la API para pasar los valores de las coordenadas de la pantalla a local, `screenToLocal`, obtendremos las coordenadas X e Y del punto con mayor facilidad.

Para este mecanismo se va a utilizar nuevamente la gestión de la contraseña mediante fichero externo. Por defecto el usuario en este sistema deberá visualizar cuatro números en el orden que desee. Por cada número que se haya observado durante la creación de la contraseña, se añadirá el identificador correspondiente al fichero, quedando así almacenada la contraseña.

Al igual que en el mecanismo de autenticación de patrón de movimiento, una vez que el usuario haya creado la contraseña y acceda a la escena de *login*, se volcará el contenido del fichero en una variable de tipo String. Además, se creará otra variable de tipo String para almacenar la contraseña introducida en la escena de *login* para que posteriormente se puedan comparar y decidir si la autenticación ha sido correcta o no.

Otras observaciones:

- Por motivos de usabilidad, la aplicación empezará a ejecutar el algoritmo una vez haya pasado un segundo desde que se accedió a la escena de contraseña. De esta manera el usuario tendrá tiempo para visualizar durante un instante el primer punto de referencia.
- Mientras se está creando la contraseña, el hilo de la aplicación guardará un tiempo de reposo para que el usuario pueda mirar de un número a otro de manera razonable y que no afecte al funcionamiento de la aplicación.
- Siempre que el usuario no esté mirando ninguno de los números o bien esté mirando las casillas grises, la aplicación seguirá recogiendo valores hasta que alguno sea válido.

3.4 Código alfanumérico

Este sistema de autenticación está asociado al uso de códigos alfanuméricos, también llamado PIT (Personal Identification Text). Esta forma de identificar al usuario consiste en establecer como contraseña una cadena que puede tener tanto dígitos como letras y es la más usada en las diferentes páginas de internet. En comparación con el código númerico, proporciona una mayor seguridad ya que el número de posibles combinaciones para generar una contraseña es muy superior. Por lo tanto, se considera una buena opción para implementar con tecnología eyetracking.

A continuación, se va a mostrar la escena de la contraseña y de *login* que se ha generado para este caso:

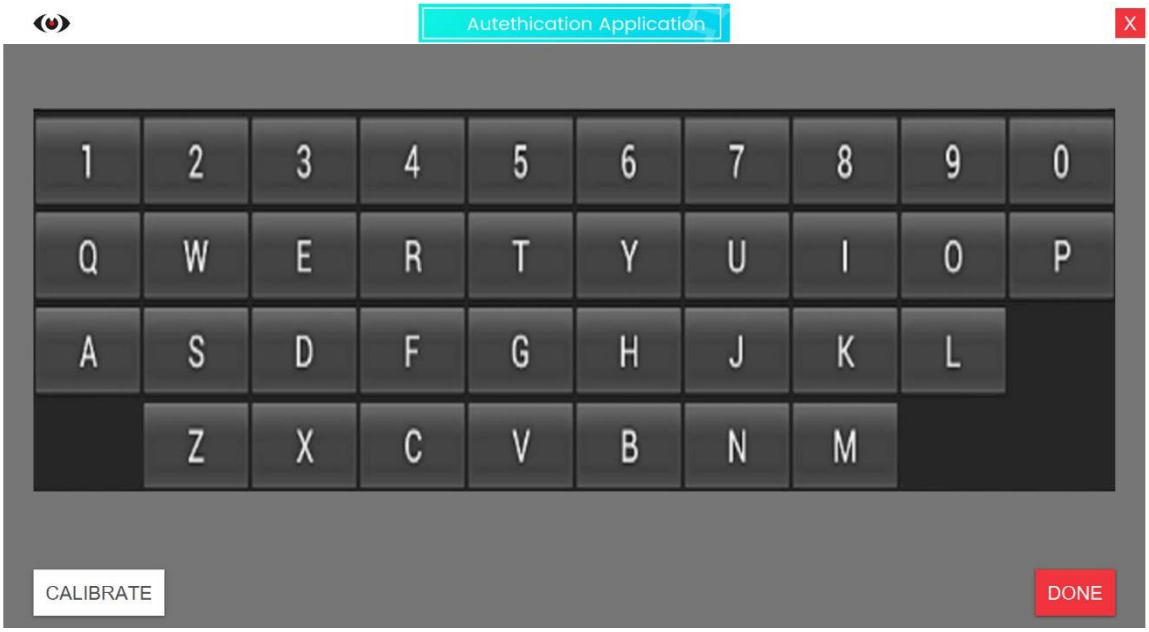


Ilustración 15: Escena password/login de código alfanumérico.

Como se puede apreciar en la imagen, además de los números, ahora se pueden visualizar letras. Tal y como se hizo en el mecanismo de código numérico, se va a trabajar con la separación de los bloques por coordenadas:

Ilustración 16: Coordenadas de código alfanumérico.

El funcionamiento de este mecanismo de autenticación es muy similar al mecanismo numérico. En este caso también se ha definido un identificador para cada uno de los bloques, teniendo un total de treinta y nueve valores. Se va a utilizar el fichero externo para almacenar la contraseña y posteriormente volcarla en una variable de tipo String.

Otra característica que se ha establecido en este mecanismo, es el número de caracteres necesarios para crear la contraseña. Se ha definido como longitud mínima de la contraseña ocho caracteres tal y como recomienda el NIST (National Institute of Standards and Technology) [30]. También se ha adaptado el código para que el tiempo de reposo generado por la aplicación entre los caracteres visualizados cuando el usuario está creando la contraseña sea lo más bajo posible y de esta manera hacer que la usabilidad de la aplicación sea la mejor posible.

3.5 Imagen fija

Se va a replicar uno de los mecanismos de autenticación más interesantes que se han encontrado para implementar con tecnología eyetracking y que pretende sustituir la autenticación mediante código PIN por la autenticación mediante el reconocimiento de una imagen [15]. En lugar de utilizar números, se va a utilizar una imagen de fondo de escena de contraseña y de *login*, preferiblemente que contenga numerosos puntos de referencia, de manera que el usuario tendrá que visualizar dentro de la imagen tantos puntos como se haya definido la longitud de la contraseña.

Para empezar a trabajar en este mecanismo, se ha establecido por defecto una longitud de contraseña de cuatro puntos de la imagen, como en el mecanismo de autenticación del código numérico. A continuación se va a mostrar la imagen seleccionada para la implementación de este mecanismo:

Ilustración 17: Escena password/login de imagen fija.

En referencia al funcionamiento del código implementado para este mecanismo, se va a trabajar directamente con las coordenadas de los puntos que van a formar la contraseña.

Una vez que el usuario haya iniciado la aplicación y se sitúe en la escena de creación de la contraseña, se van a ir almacenando las coordenadas de los puntos que el usuario vaya visualizando en un fichero externo. En esta ocasión, para almacenar los valores de la contraseña en el fichero con formato String y para facilitar una comparación posterior, los puntos se van a almacenar línea a línea. Para este mecanismo de autenticación también se aplica una pequeña parada temporal en el hilo de la aplicación entre la recogida de los valores y también un segundo inicial para visualizar la imagen.

Cuando el usuario haya creado la contraseña, accederá a la escena de autenticación. El primer paso de la aplicación en este punto es volcar toda la información del fichero a un array de tipo Double que se ha llamado `contraseña`. El siguiente paso es la recogida de los valores de los puntos que el usuario ha visualizado durante la autenticación. Estos valores se van a almacenar en otro array de tipo Double llamado `login`.

Finalmente, se realiza una comparación de los valores de los dos arrays. Siendo conscientes de que los valores son decimales, se presupone complicado que los valores de las coordenadas que forman la contraseña y los valores de las coordenadas obtenidos al

autenticar sean iguales. Para solucionar este obstáculo, se ha establecido un umbral de ochenta píxeles como margen de error a la hora de realizar la autenticación:

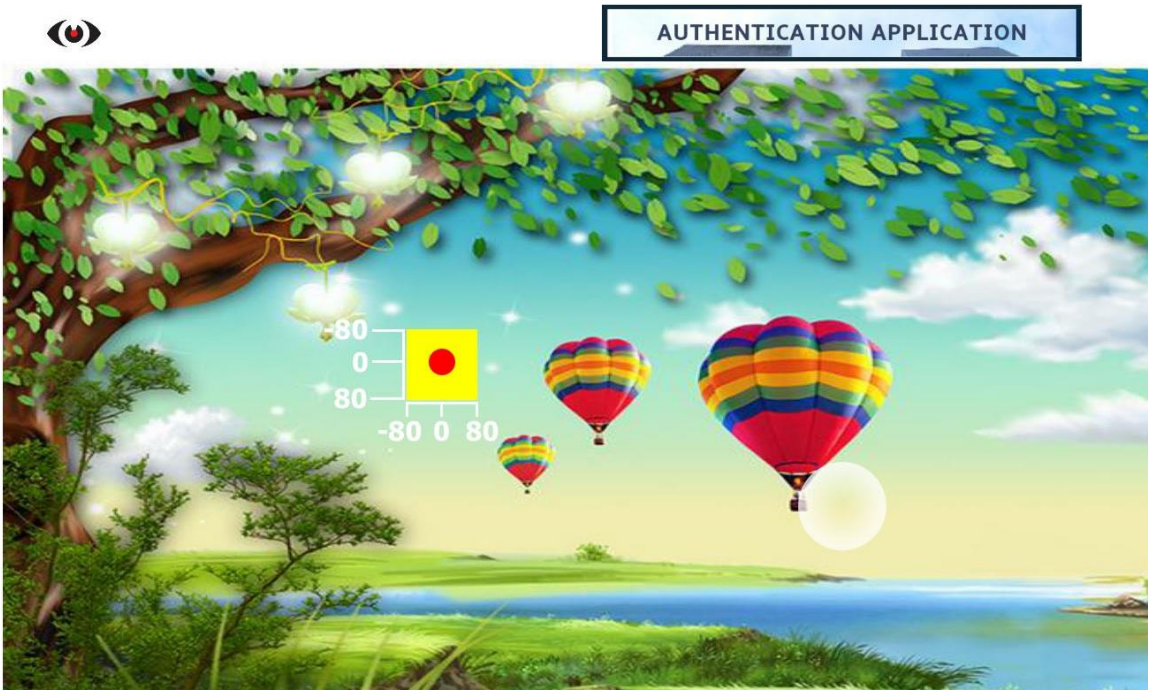


Ilustración 18: Umbral aplicado a punto de contraseña.

El circulo rojo de la imagen equivaldría al punto que el usuario visualizó durante la creación de la contraseña, mientras que la zona amarilla es la región que representa todos los valores de la autenticación que se van a considerar como válidos.

Capítulo 4

Resultados y comparación

En este cápitulo se van a desarrollar los resultados obtenidos en las pruebas realizadas en cada uno de los mecanismos de autenticación. Se ha optado por dividir las pruebas de dichos mecanismos en tres casos distintos, de contraseña más corta a contraseña más larga para poder visualizar la evolución del rendimiento de cada mecanismo de autenticación al modificar la longitud de la contraseña.

Para valorar y comparar el nivel de seguridad se va a cuantificar la fuerza de cada contraseña mediante la métrica de entropía que calcula el número de bits que representa todas las posibles contraseñas. Este valor se va a denominar TPS (Theoretical Password Spaces) y para los casos del código PIN y código alfanumérico se va a obtener mediante la siguiente formula:

$$TPS = \log_2(c^n) \text{ [31]}$$

El carácter "c" representa el número de caracteres y/o digitos disponibles y el carácter "n" representa la longitud de la contraseña.

Para el caso del patrón de movimiento, obtendremos el TPS mediante el binomio de Newton del número total de bloques en la pantalla y el número de bloques que forman la contraseña (denominado "n") [32]:

$$\text{TPS} = \log_2\binom{3\ x\ 4}{n}$$

Por último, para el mecanismo de autenticación de imagen fija, se van a tener en cuenta los puntos de interés (puntos, líneas, círculos, cruces…) y se va a calcular el número de combinaciones posibles entre ellos, obteniendo así el TPS, con la misma fórmula utilizada para los mecanismos de código PIN y código alfanumérico.

Siguiendo las pautas de un estudio llevado a cabo sobre la fortaleza de las contraseñas [31] y con el fin de poder comparar estos valores, se va a establecer el siguiente umbral en bits:

$$TPS < 15 \rightarrow nivel\ de\ seguridad\ bajo$$
$$15 < TPS < 25 \rightarrow nivel\ de\ seguridad\ medio$$
$$TPS > 25 \rightarrow nivel\ de\ seguridad\ alto$$

Las pruebas que se han realizado constan de diez muestras, las cuales van a recoger si la prueba ha sido exitosa, si ha sido incorrecta y el tiempo (expresado en segundos) que ha sido necesario para realizar la autenticación. Cuando se hayan recogido los valores de las diez muestras, se va a establecer el porcentaje de éxito, el tiempo medio y el grado de seguridad.

Una vez se hayan expuesto todos los resultados de los cuatro mecanismos, se va a realizar una comparación entre ellos y un posterior análisis.

4.1 Resultados

En este apartado se van a desglosar los resultados obtenidos para cada uno de los mecanismos de autenticación.

4.1.1 Resultados patrón de movimiento

Las longitudes de las contraseñas que se han decidido establecer para realizar las pruebas de este mecanismo de autenticación son de tres, cuatro y cinco bloques. A continuación, se muestran los resultados obtenidos:

Movimiento	Acceso correcto	Acceso incorrecto	Porcentaje de éxito	Tiempo empleado medio [s]	TPS [bits]	Nivel de seguridad
3 bloques	10	0	100%	1,46	8	bajo
4 bloques	10	0	100%	1,62	9	bajo
5 bloques	9	1	90%	1,72	10	bajo

Tabla 2: Resultados del mecanismo de autenticación de patrón de movimiento.

Como se puede apreciar, el porcentaje de éxito es muy elevado en todos los casos, lo cual resulta lógico ya que es un mecanismo de autenticación simple. Además, el tiempo medio empleado para la autenticación es muy pequeño, ya que la aplicación no se detiene en ningún momento haciendo que el proceso de autenticación sea rápido. La gran desventaja de este mecanismo de autenticación es que el nivel de seguridad que ofrece no es aceptable.

4.1.2 Resultados código PIN

Históricamente las contraseñas en forma de código PIN han tenido una longitud de cuatro dígitos. Es razonable pensar que esto es debido a que es una longitud que ofrece una cantidad de combinaciones posibles lo suficientemente grande como para que sea difícil de descifrar. Ademas suelen contar con un sistema de bloqueo si se ha errado al introducir la contraseña en cinco intentos.

Las longitudes de las contraseñas que se van a utilizar durante la realización de las pruebas son de tres, cuatro y cinco dígitos. Se van a mostrar los resultados de las pruebas correspondientes a la implementación del código PIN como mecanismo de autenticación en la Tabla 3.

Código PIN	Acceso correcto	Acceso incorrecto	Porcentaje de éxito	Tiempo medio empleado [s]	TPS [bits]	Nivel de seguridad
3 números	9	1	90%	2,64	10	bajo
4 números	8	2	80%	3,06	14	bajo
5 números	6	4	60%	3,72	17	medio

Tabla 3: Resultados del mecanismo de autenticación código PIN.

Puede observarse que el porcentaje de éxito en la autenticación disminuye cuanto más larga sea la longitud de la contraseña. Por otro lado, el nivel de seguridad del sistema solamente será aceptable cuando la longitud de la contraseña sea cinco o mayor, es decir, cuando el número de combinaciones posible sea de 100.000 (10^5) o superior. Por último, podemos ver que el tiempo medio que tarda el usuario en realizar la autenticación en todos los casos es relativamente pequeño y además la interfaz es intuitiva, por lo que la usabilidad general del sistema es aceptable.

4.1.3 Resultados código alfanumérico

Las pruebas que han sido realizadas para la autenticación mediante un código afanumérico han sido llevadas a cabo con contraseñas de ocho, nueve y diez caracteres de longitud. Se han escogido estas longitudes ya que los expertos recomiendan que la longitud mínima de una contraseña alfanumérica sea de ocho caracteres.

Estos son los resultados correspondientes a las pruebas realizadas para la autenticación mediante el código alfanumérico:

Código alfanumérico	Acceso correcto	Acceso incorrecto	Porcentaje de éxito	Tiempo empleado medio [s]	TPS [bits]	Nivel de seguridad
8 caracteres	5	5	50%	7,1	41	alto
9 caracteres	4	6	40%	7,88	47	alto
10 caracteres	3	7	30%	8,7	52	alto

Tabla 4: Resultados del mecanismo de autenticación código alfanumérico.

Como se puede apreciar en la tabla 4, el porcentaje de éxito es igual o inferior al 50% para todos los casos. Esto es debido a que la implementación de este mecanismo con tecnología *eyetracking* es compleja ya que el usuario debe visualizar varios puntos de la pantalla que en muchas ocasiones se encuentran en extremos opuestos. Esto lleva a concluir

que la usabilidad no alcanza el nivel esperado y que la experiencia de los usuarios utilizando este mecanismo de autenticación no es buena, ya que en la mitad de los casos no van a poder autenticarse al primer intento.

En cuanto al resto de parámetros, el tiempo medio de autenticación es elevado debido a la necesidad de visualizar entre ocho y diez caracteres en la pantalla. No obstante, el punto fuerte de este mecanismo es el nivel de seguridad ya que, en el peor de los casos, el número total de combinaciones posibles es de $3x10^{12}$.

4.1.4 Resultados imagen fija

En cuanto al mecanismo de autenticación mediante imagen fija, se ha establecido que la longitud de las contraseñas a probar sea de tres, cuatro y cinco puntos de la imagen.

Imagen fija	Acceso correcto	Acceso incorrecto	Porcentaje de éxito	Tiempo empleado medio [s]	TPS [bits]	Nivel de seguridad
3 puntos	10	0	100%	3,5	17	medio
4 puntos	10	0	100%	4,2	22	medio
5 puntos	8	2	80%	4,6	27	alto

Tabla 5: Resultados del mecanismo de autenticación de imagen fija.

Analizando los resultados obtenidos, se puede apreciar que es un mecanismo que ofrece una gran usabilidad porque el tiempo medio que dura la autenticación para todos los casos es adecuado a las necesidades y también porque el porcentaje de éxito refleja la consistencia del mecanismo implementado, mostrando al menos un 80% de porcentaje de éxito.

En cuanto al nivel de seguridad, se han contado 41 puntos de interés en esta imagen, resultando en 68.921 posibles combinaciones en el peor de los casos (41^3). Ya que los puntos de referencia son subjetivos, este número de combinaciones puede variar de un usuario a otro, pero en cualquier caso el nivel de seguridad que ofrece es aceptable.

4.2 Comparación

En este apartado vamos a realizar una comparación objetiva entre los mecanismos de autenticación implementados, tomando como referencia los valores obtenidos en las pruebas. Se van a comparar los diversos parámetros utilizados, primero los porcentajes de éxito, seguido del tiempo medio utilizado y por último los niveles de seguridad.

4.2.1 Porcentajes de éxito

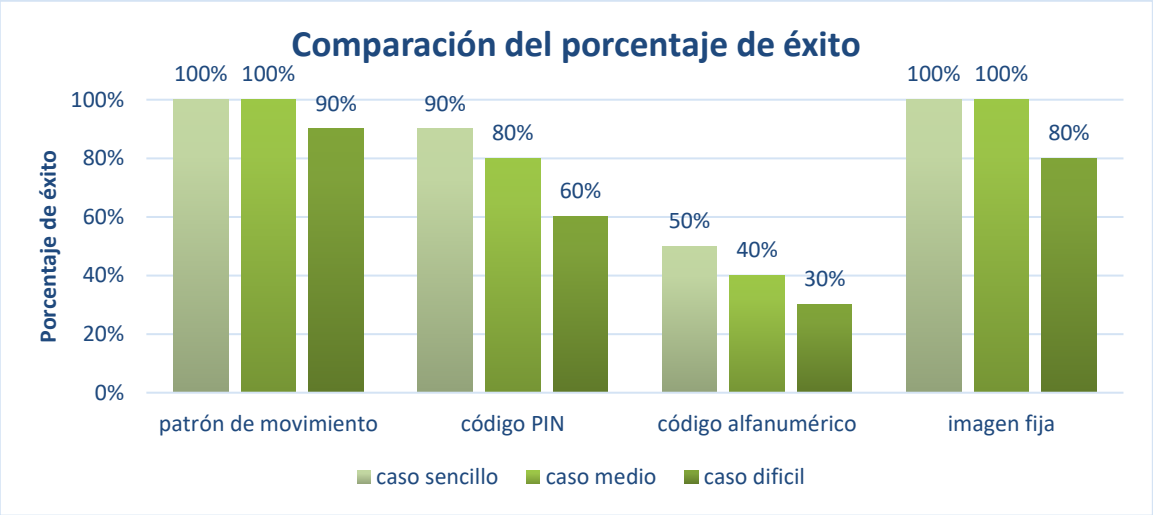


Ilustración 19: Comparación de porcentajes de éxito.

Para saber si un mecanismo de autenticación es fiable, se va a utilizar el porcentaje de éxito como referencia. Las características más importantes de una aplicación es que sea fiable y fácil de usar, por lo que, gracias al porcentaje de éxito de las pruebas realizadas, podremos identificar aquellos mecanismos que han tenido un mayor número de intentos exitosos sobre el número total de intentos, demostrando así una mejor usabilidad.

Analizando la gráfica se puede deducir que los mecanismos más fiables son el patrón de movimiento y la imagen fija. Ambos sistemas presentan un porcentaje de éxito igual o superior al 80%. Por otro lado, los mecanismos de autenticación de código PIN y código alfanumérico presentan un rango de porcentaje de éxito inferior y por tanto su fiabilidad será baja.

El motivo por el que unos mecanismos son más fiables que otros es la complejidad que presenta el funcionamiento de los mecanismos en los que el usuario tiene que girar la mirada rápidamente de un punto situado en una zona delimitada en la pantalla a otro punto situado en la otra punta del teclado virtual.

4.2.2 Tiempos medios utilizados

Otra de las características de vital importancia para los usuarios es el tiempo que hay que emplear para realizar la autenticación.

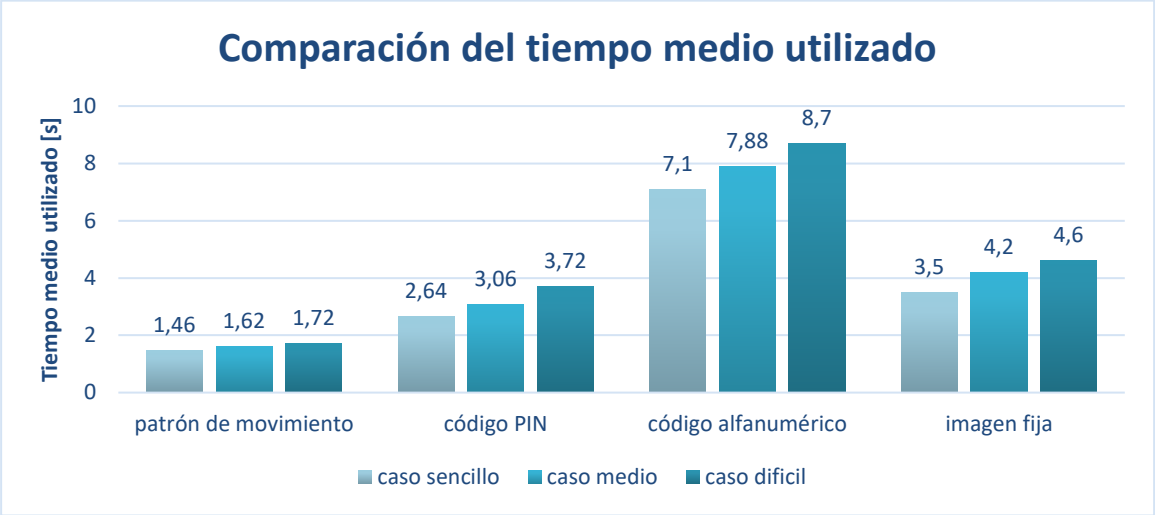


Ilustración 18: Comparación de tiempo medio utilizado.

Según los resultados obtenidos en las pruebas, cabe destacar que solamente el mecanismo de autenticación del código alfanumérico presenta un tiempo medio elevado, siendo este entre 7,1 y 8,7 segundos. La razón es evidente, se trata del mecanismo de autenticación que tiene las contraseñas de mayor longitud, por lo que la aplicación deberá trabajar durante más tiempo.

En cuanto al resto de mecanismos de autenticación, se puede apreciar que el tiempo medio utilizado es aceptable. En relación a la autenticación mediante el patrón de movimiento, destaca la pequeña cantidad de tiempo medio que aumenta de un caso a otro, del caso sencillo al caso medio aumenta 0,16 segundos y del caso medio al caso avanzado aumenta 0,10 segundos. Es sin duda el mecanismo que mejores prestaciones temporales proporciona. Los mecanismos código PIN e imagen fija tienen unos resultados de tiempo medio utilizado parecidos ya que las longitudes de las contraseñas para ambos sistemas son iguales y el tiempo que la aplicación está detenida para la recogida de valores también es similar.

4.2.3 Niveles de seguridad

Para realizar la comparación del nivel de seguridad que proporciona cada mecanismo de autenticación implementado, se han utilizado los valores del TPS de forma que si el TPS es inferior a 15 (nivel de seguridad bajo) se representará en la gráfica con un uno, si el TPS está entre 15 y 25 (nivel de seguridad medio) se representará en la gráfica con un dos y si el TPS es superior a 25 se representará en la gráfica con un tres.

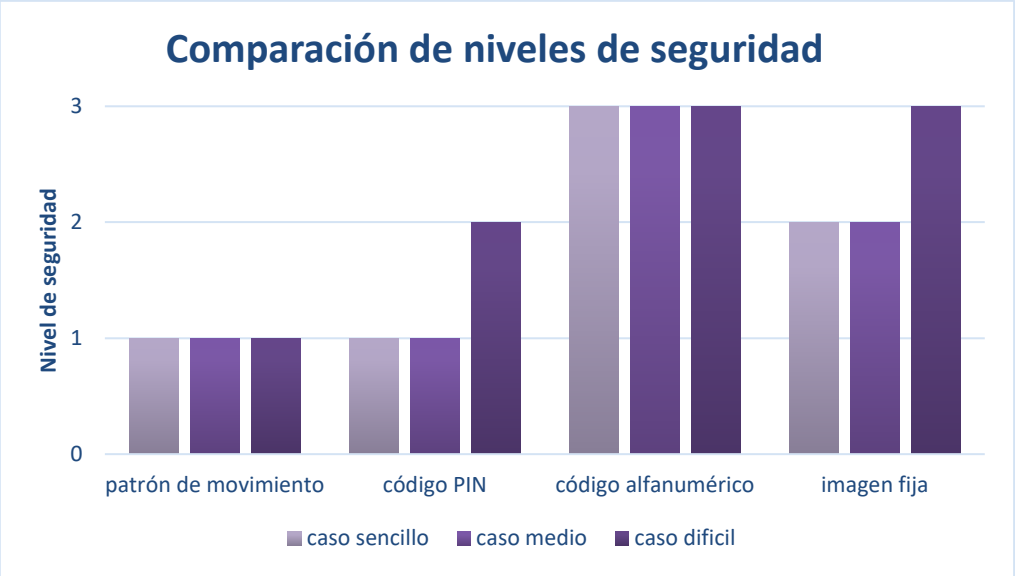


Ilustración 19: Comparación de niveles de seguridad.

Analizando la Ilustración 19 podemos ver que la autenticación realizada mediante el patrón de movimiento es la menos segura. Este mecanismo presenta mayores vulnerabilidades, ya que el número total de bloques, doce, junto al funcionamiento del mecanismo que es de bloques contiguos, hace que las posibles combinaciones no sean suficientemente altas.

En cuanto al resto de mecanismos de autenticación, el nivel de seguridad está relacionado con el número de combinaciones posibles que ofrecen. Se puede apreciar que, en el mecanismo del código alfanumérico, el nivel de seguridad es alto gracias a que la contraseña puede estar compuesta por 36 caracteres distintos sumado a que la longitud de la contraseña es de 8 a 10 caracteres. También es destacable el buen nivel de seguridad que presenta el mecanismo de la imagen fija, ya que las imágenes las puede seleccionar el usuario y puede escoger aquella imagen que tenga tantos puntos de referencia como quiera, elevando así las combinaciones posibles para formar la contraseña.

4.2.4 Comparación general

A continuación, se van a detallar las ventajas y desventajas que presentan los diferentes mecanismos de autenticación.

Mecanismo de autenticación	Ventajas	Desventajas
Patrón de movimiento	Buenas prestaciones en cuanto al tiempo medio de uso y a la usabilidad.	Bajo nivel de seguridad, demasiado intuitivo.
Código PIN	Usabilidad aceptable con un nivel de seguridad medio.	Cuantos más dígitos compongan la contraseña, el rendimiento será peor.
Código alfanumérico	Alto nivel de seguridad, el número de combinaciones posibles para la contraseña es elevado.	La usabilidad del mecanismo con esta tecnología es mala, bajo porcentaje de éxito y tiempo de uso muy grande.
Imagen fija	Mecanismo dinámico, con gran variedad de contraseñas, gran usabilidad.	Si la imagen elegida no tiene puntos de referencia evidentes, se complica la autenticación.

Tabla 6: Ventajas y desventajas de los mecanismos de autenticación.

El mecanismo de autenticación que mejores prestaciones ha proporcionado durante este estudio ha sido la imagen fija. No solo se ha demostrado que es un sistema con gran seguridad, sino que la usabilidad que presenta es muy alta, gran porcentaje de éxito y bajo tiempo medio de uso. En comparación a los demás mecanismos de autenticación, se trata de un sistema dinámico y amigable que ofrece posibilidades que ningún otro sistema puede ofrecer, ya que el escenario de la autenticación es variable.

Por el contrario, el mecanismo de autenticación que peores prestaciones ha demostrado tener ha sido el código alfanumérico. La interacción con el dispositivo a la hora de autenticarse se hace demasiado larga, lo que conlleva a que la relación usabilidad-seguridad de este mecanismo esté desbalanceada, ya que presenta una gran seguridad, pero no tiene un funcionamiento estable.

Capítulo 5

Planificación del trabajo y presupuesto

En el presente capítulo se va a proceder a la explicación de las fases en las que se ha completado el proyecto. Para ilustrar la planificación y la distribución temporal de las fases se ha realizado un diagrama de Gantt donde se pueden visualizar las distintas tareas realizadas y el tiempo de dedicación previsto para cada una de ellas. También se van a especificar todos los costes relacionados con la realización del proyecto.

5.1 Planificación del trabajo

A lo largo de este apartado se van a explicar las fases en las que se ha dividido la realización del trabajo de fin de grado.

5.1.1 Definición de tareas

La elaboración del presente proyecto se ha dividido en tres fases, que se enumeran a continuación:

- Fase 1: Planificación.

Para comenzar la realización del trabajo primero es necesario realizar un estudio previo acerca del *hardware* y del *software* que se va a utilizar, de los mecanismos de autenticación que se pueden implementar y también es necesario hacer la definición de los objetivos que se desean alcanzar.

  - Planteamiento del trabajo. Estudio de los objetivos requeridos, la finalidad de estos y la organización para alcanzarlos.
  - Adquirir el conocimiento suficiente para trabajar con el dispositivo *eyetracker* y sus características.
  - Selección de herramientas a utilizar para el desarrollo del proyecto.
  - Estudio de los mecanismos de autenticación a implementar en el proyecto.

- Fase 2: Ejecución.

A continuación, una vez formados en el uso del dispositivo *eyetracker*, se va a realizar la implementación de los mecanismos de autenticación que han sido

seleccionados para la realización del proyecto. Se va a desarrollar el código y se va a hacer un análisis de los resultados obtenidos con los que se podrá realizar una comparación posteriormente.

- Instalación del entorno de trabajo, Eclipse y las librerías de JavaFX.
- Desarrollo del código.
- Realización de pruebas. Una vez desarrollado el código se realizan los experimentos.
- Análisis de los resultados de los mecanismos de autenticación.
- Comparación de los resultados.

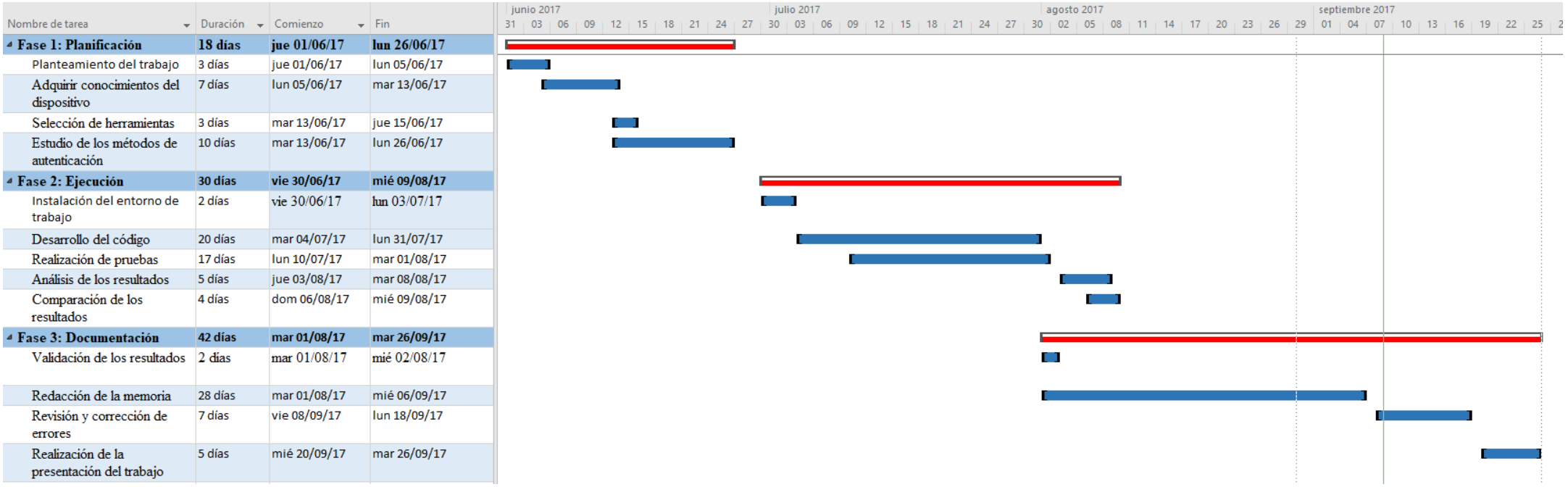
- Fase 3: Documentación.

Tras la validación de los resultados obtenidos, se va a realizar la redacción de la presente memoria, así como la presentación del proyecto.

- Validación de los resultados obtenidos.
- Redacción de la memoria.
- Revisión y corrección de errores.
- Realización de la presentación del trabajo.

5.1.2 Diagrama de Gantt

Una vez definidas las fases del proyecto, se va a realizar el diagrama de Gantt para reflejar el tiempo estimado asociado a cada una de las tareas que componen el proyecto. Será necesario indicar en él las fechas de inicio y de fin estimadas. Con esta planificación se tendrá un control temporal y del progreso del proyecto.



Nombre de tarea	Duración	Comienzo	Fin
Fase 1: Planificación	**18 días**	**jue 01/06/17**	**lun 26/06/17**
Planteamiento del trabajo	3 días	jue 01/06/17	lun 05/06/17
Adquirir conocimientos del dispositivo	7 días	lun 05/06/17	mar 13/06/17
Selección de herramientas	3 días	mar 13/06/17	jue 15/06/17
Estudio de los métodos de autenticación	10 días	mar 13/06/17	lun 26/06/17
Fase 2: Ejecución	**30 días**	**vie 30/06/17**	**mié 09/08/17**
Instalación del entorno de trabajo	2 días	vie 30/06/17	lun 03/07/17
Desarrollo del código	20 días	mar 04/07/17	lun 31/07/17
Realización de pruebas	17 días	lun 10/07/17	mar 01/08/17
Análisis de los resultados	5 días	jue 03/08/17	mar 08/08/17
Comparación de los resultados	4 días	dom 06/08/17	mié 09/08/17
Fase 3: Documentación	**42 días**	**mar 01/08/17**	**mar 26/09/17**
Validación de los resultados	2 días	mar 01/08/17	mié 02/08/17
Redacción de la memoria	28 días	mar 01/08/17	mié 06/09/17
Revisión y corrección de errores	7 días	vie 08/09/17	lun 18/09/17
Realización de la presentación del trabajo	5 días	mié 20/09/17	mar 26/09/17

Ilustración 20: Diagrama de Gantt.

5.2 Presupuesto

Se va a proceder a realizar el desglose de los costes, tanto materiales como de personal, asociados a la realización de este proyecto.

5.2.1 Costes materiales

En este apartado se va realizar el cálculo de los costes asociados a la adquisición de los materiales físicos necesarios para realizar el proyecto. Las licencias de los programas que se han tenido que adquirir sin ser gratuitos en el proyecto también se van a incluir en el coste material.

CONCEPTO	CANTIDAD	COSTE (€)	TIEMPO DE USO (MESES)	VIDA ÚTIL (MESES)	COSTE imputable (€)
Intel Core i5-7500 CPU @ 3.40 GHz	1	933,5	4	48	77,79
Licencia Microsoft Office 2016 Personal	1	69	3	12	17,25
Dispositivo de seguimiento de ojos + SDK	1	90	4	24	15
Gastos de electricidad	-	50	-	-	50
Conexión a Internet 300Mbps	-	160	-	-	160
				COSTE TOTAL	320,04

Tabla 7: Desglose de los costes materiales del proyecto.

Los costes imputables materiales se van a obtener mediante la siguiente fórmula:

$$Coste\ imputable = \frac{(Tiempo\ de\ uso\ x\ Coste)}{Vida\ útil}$$

5.2.2 Costes de personal

Para calcular los **costes de personal** se van a tener en cuenta a las dos personas implicadas en el proyecto, el autor del Trabajo de Fin de Grado y las doctoras Tutoras del mismo.

A continuación, se muestra una tabla basada en la categoría, el tiempo invertido en la realización (en horas) y el sueldo (en Euros) de la persona.

NOMBRE Y APELLIDOS	CATEGORÍA	HORAS DE TRABAJO (h)	COSTE (€/h)	TOTAL (€)
Florina Almenarez Mendoza	Dra. Ingeniera	40	45	1.800
Patricia Arias Cabarcos	Dra. Ingeniera	40	45	1.800
Sergio Cerrada Lerena	Graduado	360	15	5.400
			TOTAL	9.000

Tabla 8: Desglose de los costes de personal del proyecto.

5.2.3 Costes totales

Finalmente, se va a realizar una tabla para representar el **coste total** del Proyecto con la suma de los costes materiales, personales e impuestos.

CONCEPTO	COSTE (€)
Material	320,04
Personal	9.000
TOTAL	9.320,04

Tabla 9: Costes totales del proyecto.

El coste total de la realización completa del proyecto es de **9.320,04 Euros**.

Capítulo 6

Conclusions and future improvements

6.1 Conclusions

This Final Project consisted on the implementation and comparisson of several authentication mechanisms based on the eyetracking technology. A research was done in order to select the authentication mechanisms that were going to be part of this project.

First of all, it was necessary to know which tools were necessary to design the authentication mechanisms solutions. This obstacle was overcome researching the system requirements. Based on the autor's programming skills, the decission of using Java as the programming language was made. In addition to this, a research of the Java libraries needed as well as the device's API was done to make the code work easier.

Once the implementation of the authentication mechanisms and the testing were done, it could be seen that the application worked properly. This means that the implemented and tested systems are a real alternative to the traditional authentications mechanisms. They not only provide a higher security level but also the usability is a step forward compared to the actual mechanisms.

As seen on the results of the tests, it can be stated that the goal of this project has been achieved as the implementation of the authentication mechanisms were successful and the comparison between them was done.

When the project has been finished and the knowledge of the eyetracking technology has been adquired in depth, it has been observed the possibility of improvements to the designed solutions.

6.2 Future improvements

To finish this bachelor thesis we have created a list of possible future developments on the same topic:

- **Addition of algorithms**. Development of other algorithms to compare with the already done ones.
    - Text reading[33]
    - Object PassTiles [34]
    - Image PassTiles [34]
- **Optimization of the already developed algorithms.** Development of code blocks that improve the usability and security of the implemented authentication mechanisms.
    - Save the password into a secure storage
    - Encrypt the password
    - Improve the eye gaze points management
- **Implement into others operating systems**. Development of the project on an OS compatible with the SDK, as Linux, Mac or Android.
- **Use different languages**. The scenes created for this project has been done in English but to increase the usability of the application, it might be necessary to change the language, making it more accesible to the society.
- **Perform the study on several users**. Including more exhaustive usability studies.
- **Add more parameters to the comparison.** Environmental conditions (light, position, distance…)

Referencias

[1] Appleinsider. “Apple buys German eye tracking firm SensoMotoric Instruments”. Disponible online: http://appleinsider.com/articles/17/06/26/apple-buys-german-eye-tracking-firm-sensomotoric-instruments (Último acceso: septiembre 2017)

[2] Deloitte. “One billion smartphone upgrades”. https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Technology-Media-Telecommunications/gx-tmt-pred15-one-billion-smartphone.pdf (Último acceso: septiembre 2017)

[3] Statista. “Number of smartphone users worldwide from 2014 to 2020 (in billions)”. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ (Último acceso: septiembre 2017)

[4] Tom Le Bras. “[INFOGRAPHIC] Online Overload – It’s Worse Than You Thought”. Disponible online: https://blog.dashlane.com/infographic-online-overload-its-worse-than-you-thought/ (Último acceso: septiembre 2017)

[5] NIST. “Report: Authentication Diary Study”. Disponible online: http://discovery.ucl.ac.uk/1434744/1/Steves%20et%20al.%20-%202014%20-%20Report%20Authentication%20Diary%20Study.pdf (Último acceso: septiembre 2017)

[6] Samuel M. “Does one password reset cost your company $7 or $70 every time? The password is... Disponible online: http://www.sparkhound.com/learn/blog/does-one-password-reset-cost-your-company-7-or-70-every-time-the-password-is (Último acceso: septiembre 2017)

[7] TheEyeTribe. “Preliminary tech specs”. http://showstoppers-mwc.vporoom.com/TheEyeTribe/download/Product+Sheet+Pro+Tracker.pdf (Último acceso: septiembre 2017)

[8] Edwin Dalmaijer. “The EyeTribe stops and that stinks”. https://www.pygaze.org/2016/12/the-eyetribe-stops-and-that-stinks/ (Último acceso: septiembre 2017)

[9] Oracle. “JavaFX: Getting Started with JavaFX”. http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784 (Último acceso: septiembre 2017)

[10] Eduardo Canelles. “¿Qué es el “Eye Tracking” y para qué nos sirve?”. Disponible online: http://www.solucionesc2.com/que-es-el-eye-tracking-y-para-que-nos-sirve/ (Último acceso: septiembre 2017)

[11] Tobii. “The Essentials of Eye Tracking”. Disponible online: https://www.tobii.com/tech/technology/what-is-eye-tracking/ (Último acceso: septiembre 2017)

[12] Eyetracking. “Case studies: advertising & sponsorship”. Disponible online: http://www.eyetracking.com/Case-Studies/Advertising-Sponsorship (Último acceso: septiembre 2017)

[13] Ergoestudio. “Diferentes aplicaciones de la tecnología Eye Tracking”. Disponible online: http://www.ergoestudio.com/articulos/articulos/diferentes_aplicaciones_et.php (Último acceso: septiembre 2017)

[14] Imotions. “Top 8 Eye Tracking Applications in Research”. Disponible online: https://imotions.com/blog/top-8-applications-eye-tracking-research/ (Último acceso: septiembre 2017)

[15] A. Yun, Zhang, B. Zheru, Chi, C. Dagan, Feng, "An analysis of eye movement based authentication systems", ICMET 2011.
[16] Tobii Pro X3–120 Eye Tracker. Disponible online: https://www.tobiipro.com/siteassets/tobii-pro/product-descriptions/tobii-pro-x3-120-product-description.pdf/?v=1.0.7 (Último acceso: septiembre 2017)
[17] Eyetech Digital Systems. "Technical Specifications for the AEye eye tracker and eye tracking api". Disponible online: https://www.eyetechds.com/board-level-oem.html (Último acceso: septiembre 2017)
[18] Neurotechnology. "SentiGaze 1.1 SDK". Disponible online: http://download.neurotechnology.com/SentiGaze_SDK_Documentation.pdf (Último acceso: septiembre 2017)
[19] GP3 Eye Tracker. Disponible online: https://www.gazept.com/product/gazepoint-gp3-eye-tracker/ (Último acceso: septiembre 2017)
[20] Pupil docs. Disponible online: https://docs.pupil-labs.com/#developer-docs (Último acceso: septiembre 2017)
[21] "Java Development Kit". Disponible online: https://en.wikipedia.org/wiki/Java_Development_Kit (Último acceso: septiembre 2017)
[22] Techopedia. "Java Runtime Environment (JRE)". Disponible online: https://www.techopedia.com/definition/5442/java-runtime-environment-jre (Último acceso: septiembre 2017)
[23] JavaFXTutorials. "What is JavaFX?". Disponible online: http://www.javafxtutorials.com/whatisjavafx/ (Último acceso: septiembre 2017)
[24] Päivi Majaranta, Poika Isokoski, Oleg Špakov, Roope Raisamo. "Haptic Feedback in Eye Typing". Disponible online: https://www.researchgate.net/publication/290515048_Haptic_Feedback_in_Eye_Typing (Último acceso: septiembre 2017)
[25] Anthony Maeder, Sridharan Subramanian. "Gaze based user authentication for personal computer applications". Disponible online: https://www.researchgate.net/publication/224614318_Gaze_based_user_authentication_for_personal_computer_applications (Último acceso: septiembre 2017)
[26] D. Maltoni, & A. Jain (Eds.). "Biometric Authentication: ECCV 2004 International Workshop, BioAW 2004", Prague, Czech Republic, May 15, 2004, Proceedings. Vol. 3087. Springer Science & Business Media, 2004.
[27] Ley Orgánica de Protección de Datos de Carácter Personal. Disponible online: http://www.boe.es/buscar/act.php?id=BOE-A-1999-23750 (Último acceso: septiembre 2017)
[28] Patrik Matell. "Eye Tribe or Tobii, an Eye tracking comparison". Disponible online: https://conversionista.se/en/eye-tribe-vs-tobii/ (Último acceso: septiembre 2017)
[29] "Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX". Disponible online : http://www.oracle.com/technetwork/java/javase/terms/license/index.html (Último acceso: septiembre 2017)
[30] Diego Poza. "Don't Pass on the New NIST Password Guidelines". Disponible online: https://auth0.com/blog/dont-pass-on-the-new-nist-password-guidelines/ (Último acceso: septiembre 2017)
[31] Alain Forget, Sonia Chiasson, Robert Biddle. "Choose Your Own Authentication". Disponible online: https://pdfs.semanticscholar.org/fe72/fe3838dfa4bcd793a1b79010922bcedd11ac.pdf (Último acceso: septiembre 2017)

[32] Freyr Sævarsson. “Combinatorics in Pattern-Based Graphical Passwords”. Disponible online: https://www.kth.se/polopoly_fs/1.364989!/Menu/general/column-content/attachment/Saevarsson.pdf (Último acceso: septiembre 2017)
[33] Roman Bednarik, Tomi Kinnunen, Andrei Mihaila, Pasi Fränti. “Eye-Movements as a Biometric”. Disponible online: http://cs.joensuu.fi/pages/tkinnu/webpage/pdf/EyeBiometrics_SCIA2005.pdf (Último acceso: septiembre 2017)
[34] Elizabeth Stobert, Robert Biddle. “Memory Retrieval and Graphical Passwords”. Disponible online: http://hotsoft.carleton.ca/~estobert/papers/soups2013_estobert.pdf (Último acceso: septiembre 2017)
[35] Imagen del dispositivo The Eye Tribe. Disponible online: https://www.neurolize.com/DeviceDetails.aspx?DeviceGuid=dc9f9d9f-a36d-493b-9cfe-c7219c89be14 (Último acceso: septiembre 2017)
[36] Imagen con texto “bienvenidos” utilizada en la escena principal. Disponible online: http://www.liceomediterraneo.com/8407-2/ (Último acceso: septiembre 2017)
[37] Imagen del teclado numérico. Disponible online: http://www.arumeinformatica.es/blog/ios-ocultar-teclado-numerico-o-decimal-number-or-decimal-pad/ (Último acceso: septiembre 2017)

ANEXO I: Introducción (castellano)

Es un hecho conocido que el número de usuarios que poseen un dispositivo electronico aumenta cada año considerablemente. Además, se está tendiendo a almacenar cada vez una mayor cantidad de información personal en dichos dispositivos como contraseñas, contactos, reunions o imagenes, lo que significa una cantidad importante de información de valor para los atacantes.

Es debido a esto que la seguridad de los dispositivos es de vital importancia. Pero la seguridad está compuesta por varios niveles, y en este caso, vamos a tratar con el nivel más visible de cara al usuario, la autenticación para acceder al dispositivo.

1.1 Objetivos y motivaciones

Los mecanismos de autenticación más utilizados durante los últimos años, el acceso mediante código PIN y el acceso estableciendo un patrón de movimiento de los dedos, no se adaptan a las necesidades emergentes ya que estos mecanismos son vulnerables a varios ataques conocidos. Los usuarios están cada vez más preocupados acerca de la seguridad de su autenticación, queriendo proteger toda la información personal que almacenan en sus dispositivos.

La protección de los datos es un factor muy importante en la actualidad para las empresas desarrolladoras, ya que la cantidad de dispositivos y sistemas que requieren autenticación continúa creciendo. Además, es necesario que los mecanismos de autenticación tengan la mayor usabilidad posible porque los usuarios prefieren sistemas que sean fáciles de usar y de entender. Vivimos en la era digital y es crucial que la seguridad de los dispositivos en los que almacenamos la información sea lo más fuerte posible a la vez que sean sencillos de utilizar.

Pero, ¿cómo pueden ser más seguros y más usables los mecanismos de autenticación? Existen dos caminos principales que están siguiendo las compañías y los desarrolladores para lograr una autenticación más segura para los usuarios:

- La **Autenticación Multifactor (MFA)** es una forma en la que el usuario solamente puede acceder al dispositivo una vez que ha podido ofrecer dos o más pruebas diferentes de que es el dueño del dispositivo. Estas pruebas pueden ser una contraseña, una contraseña secundaria o un certificado digital.

- Acceso **basado en parámetros biométricos,** que consiste en usar características morfológicas del usuario para decidir si puede acceder o no al dispositivo. En relación a esto, durante este proyecto, se van a implementar diferentes mecanismos de autenticación basados en la mirada de los ojos.

Hasta donde se puede saber, ya existen departamentos[2] de grandes compañías que están investigando en mecanismos de autenticación basados en la tecnología eyetracking. Esto es una clara señal de que la autenticación usando mecanismos más avanzados es ya una realidad y pronto dejará atrás los mecanismos de autenticación tradicionales.

Una vez que las propuestas de las implementaciones hayan sido realizadas, se realizará una comparación exhaustiva entre los distintos mecanismos de autenticación, teniendo en cuenta diferentes criterios como puede ser la usabilidad, la funcionalidad o el nivel de seguridad.

El objetivo general de este proyecto es realizar la implementación de varios mecanismos de autenticación basados en la tecnología *eyetracking* para poder demostrar que existen técnicas de autenticación con una usabilidad fiable, más seguras que las actuales. Para ello, se han establecido los siguientes objetivos:

1. Realizar una **investigación** de los mecanismos de autenticación disponibles que encajen en este proyecto. Se realizará un análisis de aquellos mecanismos que sean propicios como soluciones al problema.
2. Alcanzar un **conocimiento elevado** del dispositivo *eyetracker* y de las librerías que facilitan la conexión entre el dispositivo y el desarrollador.
3. **Implementar** los mecanismos de autenticación. Para conseguir una buena usabilidad, el código debe ser claro y con sentido.
4. Una vez que la aplicación ha sido creada y funciona correctamente, se van a realizar las pruebas. Se va a realizar la autenticación en diez intentos para cada caso y se van a anotar los **resultados** finales de cada test: porcentaje de éxito, tiempo medio de uso, nivel de seguridad…
5. Basandose en los resultados, se realizará una comparación entre los mecanismos de autenticación.

1.2 Contexto socioeconómico

De acuerdo con varios estudios, la predicción de la cantidad de dispositivos electrónicos que van a ser vendidos durante el próximo año va a ser mayor, tal y como ha ido pasando de un año a otro sucesivamente [2]:

[2] Por ejemplo, Apple ha adquirido recientemente una empresa relacionada con la tecnología *eyetracking* llamada SensoMotoric Instruments (SMI) y han dejado entrever la posibilidad de utilizar esta tecnologia en futuros dispositivos de la marca [1].

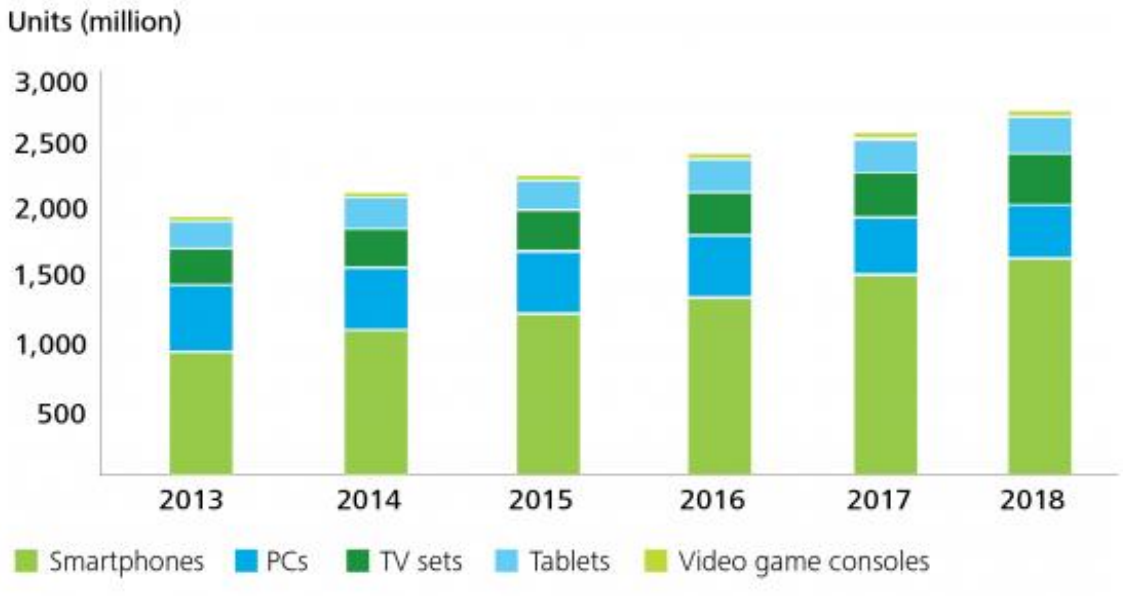


Ilustración 21: Ventas globales combinadas de PC, smartphones, tablets, televisores y videconsolas [2].

Como podemos apreciar en la imagen, los dispositivos que más se compran anualmente son los teléfonos inteligentes, los cuales tienen también el mayor aumento de ventas de un año para otro. Este incremento se debe al hecho de que este dispositivo electrónico ofrece un rango más amplio de precios, haciendo que sea accesible para toda la población. Actualmente hay 2,32 billones de usuarios de teléfonos inteligentes y la cifra que se espera alcanzar en el año 2020 es de 2,87 billones [3].

Se puede deducir fácilmente que los ingresos correspondientes al sector tecnológico durante los próximos años van a ser importantes. Por lo tanto, los desarrolladores van a tener más recursos para realizar avances en todo el sector tecnológico, incluyendo los mecanismos de autenticación. Además, esto supondrá un impacto positivo en la sociedad ya que las empresas tendrán más recursos para contratar a personas más cualificadas para el desarrollo tecnológico.

El progreso tecnológico es inevitable, ya que el uso de los dispositivos se ha convertido en una necesidad para sentirse conectado con el resto de la sociedad. Los usuarios de los teléfonos inteligentes están conectados desde que se despiertan por la mañana hasta el final del día. Sabiendo esto, la cantidad de empresas dedicadas a la tecnología mundial incrementará, así como el número de usuarios, lo cual llevará a un impacto económico en las regiones donde se asenten. Una parte del desarrollo de nuevas tecnologías tiene como objetivo hacer nuestras vidas más sencillas y más seguras.

Las ventajas están claras, pero también hay que hablar de las desventajas. Ningún tipo de dependencia es beneficioso para los miembros de la sociedad. Las personas se convierten en usuarios de teléfonos inteligentes cada vez más jovenes porque tienen como ejemplo a sus padres, quienes utilizan dispositivos electrónicos asiduamente. El uso de

dispositivos electrónicos puede llegar a ser peligroso porque cuando se comparte contenido multimedia, como fotos o videos, no se sabe con total certeza quien puede llegar a visualizar ese contenido. Los usuarios más jovenes puede que no sean conscientes del peligro que conlleva. A pesar de esto, las empresas tienden a fabricar productos que encajen con el perfil de los usuarios jovenes, animándoles a comprarlo y expandiendo así su mercado.

El usuario medio posee aproximadamente noventa cuentas *online* lo que significa que, en el peor de los casos, el usuario debe recordar noventa contraseñas. Poseer tantas cuentas lleva a los usuarios a utilizar una misma contraseña para varias cuentas, lo que compromete su seguridad debido a que, si un atacante conoce una contraseña utilizada en varios sistemas, podrá acceder a varios datos personales [29]. Esto puede causar lo que se conoce como "fatiga de contraseña", que puede ser descrito como el estado de estrés y de fatiga experimentado por los usuarios de la tecnología que se ven desbordados por la demanda de su tiempo, energía y memoria que implica utilizar los dispositivos [30].

De media, las peticiones de reiniciar contraseñas representan entre el 20% y el 50% de las llamadas a soporte en una empresa, lo que lleva a una bajada de productividad ya que resetear contraseñas provoca una pérdida de tiempo y por tanto de dinero. Según una investigación publicada por Forrester, reiniciar una única contraseña cuesta en torno a setenta dólares [31].

El desarrollo de los mecanismos de autenticación debe tender a métodos que sean más naturales o que interfieran mínimamente en la principal tarea del usuario, que es usar el servicio o el dispositivo al que quieren acceder. Además, no solo tendría un gran impacto en la salud del usuario, sino que también tendría un gran impacto económico.

1.3 Medios empleados

En relación a los recursos que han sido utilizados durante las diferentes fases de la creación del proyecto, se van a detallar las herramientas utilizadas y el papel que han tenido.

1.3.1 Hardware

Para realizar este proyecto, se va a utilizer el dispositivo *The Eye Tribe EyeTracker* fabricado por la compañía *The Eye Tribe*.

Una de las grandes ventajas que tiene utilizar este dispositivo, es el bajo precio que tiene y que hace que sea un recurso potencialmente interesante de cara a tareas de investigación. Además, el dispositivo proporciona características que aseguran un gran rendimiento [4]:

- Tasa de muestreo: 30 Hz a 75 Hz
- Precisión: 0.5° – 1°
- Latencia: < 1.6 ms
- Calibración: 6, 9, 12 puntos
- Rango de trabajo: 45cm - 75cm
- Área de seguimiento: 45cm x 30cm
- Tamaño de pantalla: Hasta 24”
- SDKs en Java, C++ y C#
- Interfaz: USB3.0 tipo C

Ilustración 22: The Eye Tribe EyeTracker.

Desafortunadamente, la empresa hizo público el 16 de diciembre de 2016 que iban a dejar de desarrollar estos dispositivos ya que habían decidido avanzar en una dirección diferente con su tecnología [5].

A pesar de esto, aún es posible adquirir este dispositivo a través de otros canales con el inconveniente de que el soporte oficial ya no va a ser una posibilidad. No obstante, la comunidad de desarrolladores va a seguir involucrada activamente, creando proyectos y resolviendo errores.

1.3.2 Software

El sistema operativo que se ha utilizado ha sido Microsoft Windows, específicamente la version gratuita de **Windows 10 Education** que proporciona la universidad Carlos III a los estudiantes.

También se ha utilizado **Microsoft Word 2016** para redactar la memoria. Está considerado como el mejor procesador de texto mundial. Se puede decir que es la herramienta de escritura más completa y potente del Mercado y siempre que sea necesario trabajar con documentos digitales, es una herramienta de gran utilidad.

Para poder realizar los diagramas de flujo de los algoritmos diseñados de los mecanismos de autenticación, se ha utilizado **Microsoft Visio 2013**.

1.3.2.1 Java

Java es un lenguaje de programación y una plataforma de computación que fue publicado por Sun Microsystems en 1995. En este proyecto, se va a utilizar la version 8 de Java para programar y ejecutar las aplicaciones creadas para cada mecanismo de autenticación.

Se ha utilizado Java Development Kit (JDK) versión 8u121 como el entorno de desarrollo para la creación de aplicaciones y componentes que utilizan el lenguaje de programación Java. El JDK incluye herramientas de gran utilidad para realizar tests y para solucionar errores de manera rápida.

Java es la herramienta más utilizada para el desarrollo aplicaciones y programas debido a numerosos factores:

- Es práctica: ha sido diseñada de forma que permita a los desarrolladores realizar su trabajo con la menor cantidad de complicaciones, permitiendo a otros desarrolladores coger el código pasado el tiempo y entender qué es lo que supuestamente debería hacer.

- Compatible con versiones futuras: el código escrito en una version de Java continuará siendo ejecutable en nuevas versiones de Java sin tener que realizar ninguna operación adicional.

- Java asegura escalabilidad, buen rendimiento y confianza.

1.3.2.2 The Eye Tribe SDK

La versión del Software Development Kit correspondiente al dispositivo The Eye Tribe EyeTracker que se ha utilizado para realizar este proyecto es la 0.9.77. Esta versión no solo proporciona una API muy completa, sino que también brinda la posibilidad de utilizar un *software* de ejemplo para que el desarrollador lo pueda usar de referencia.

Una vez que el SDK ha sido instalado, hay dos aplicaciones que pueden ser inicializadas para saber si todo está correcto. La primera de ellas se llama EyeTribe Server y permite arrancar el servidor del dispositivo. Se intenta establecer la conexión con el dispositivo y una vez que se consigue establecer, muestra por consola las trazas del estado del servidor. La segunda aplicación se llama EyeTribe UI y está hecha para mostrar las instrucciones de uso del dispositivo y una vez visualizadas, permite realizar al usuario una calibración.

Este SDK está disponible para el lenguaje de programación Java, para C ++ y para el lenguaje de programación C. Gracias a este SDK, el servidor puede gestionar las coordenadas de la pantalla a la que el usuario está mirando.

1.3.2.3 Eclipse

Para implementar el código de las aplicaciones Java, se ha utilizado Eclipse como el integrador del entorno de desarrollo (IDE), el cual es el *software* donde el código Java va a ser escrito y probado. Se ha elegido Eclipse porque este está escrito en Java y por su facilidad de uso para el desarrollo de aplicaciones Java. En este proyecto en particular, se va a utilizar la version Neon.3.

Una de las principales razones para utilizar Eclipse como el integrador del entorno de desarrollo es por las facilidades que proporciona al usuario como la arquitectura de ficheros del proyecto, la asistencia de solución de errores mientras se escribe el código, completar una parte del código o la descripción de los mecanismos disponibles en el momento.

En este proyecto es necesario descargar e instalar una extension llamada JavaFX dentro de la plataforma de Eclipse. JavaFX proporciona un conjunto de librerías de caracter gráfico y de contenido visual que permite al desarrollador diseñar, crear, probar, corregir y desplegar aplicaciones [6].

1.3.2.4 JavaFX Scene Builder 2.0

Como se ha mencionado previamente, JavaFX ofrece al desarrollador la oportunidad de trabajar con paquetes gráficos de tal forma que pueda diseñar diferentes escenas dentro de la aplicación.

Gracias a JavaFX Scene Builder, el trabajo de diseñar y crear las escenas usadas en la aplicación es mucho más sencillo. JavaFX Scene Builder es un entorno visual en el que el desarrollador puede generar interfaces de usuario y es el propio *software* el que genera el código correspondiente en un fichero FXML donde la escena queda almacenada.

1.3.2.5 Librerías

Para que el proyecto pueda salir adelante, Eclipse proporciona algunas librerías internas llamadas “JRE System Library” entre las cuales se pueden encontrar librerías de JavaFX interesantes.

Adicionalmente, es necesario añadir al proyecto una librería externa llamada EyeTribeJavaFx, la cual include numerosos métodos que van a hacer el trabajo más asequible. Esta librería se puede encontrar facilmente en la página official de The Eye Tribe.

1.4 Estructura de la memoria

Este documento contiene seis capítulos donde se van a incluir todos los pasos seguidos para alcanzar el objetivo de este proyecto, la implementación de varios mecanismos de autenticación, creándolos como aplicaciones de Java, y posteriormente realizando una comparación entre dichos mecanismos utilizando diferentes parámetros. Una breve introducción de cada capítulo se ofrece a continuación:

- Capítulo 1: Introducción. Motivación y objetivos del proyecto, el contexto socioeconómico, los recursos utilizados y la estructura de la memoria.

- Capítulo 2: Estado del arte. Una explicación de los mecanismos que van a ser implementados y de las tecnologías utilizadas.

- Capítulo 3: Descripción general de cómo se han implementado los mecanismos y las particularidades de cada uno.

- Capítulo 4: Resultados y comparaciones de los mecanismos implementados.

- Capítulo 5: Planificación del proyecto y presupuestos. Descripción de las fechas de las diferentes fases del desarrollo del proyecto y el coste de los recursos.

- Capítulo 6: Conclusión. Análisis y conclusiones obtenidas del proyecto. Líneas de futuro.

ANEXO II: Conclusiones y líneas futuras (Castellano)

6.1 Conclusiones

Este Trabjo de Fin de Grado ha consistido en la implementación y comparación de diferentes técnicas de autenticación basadas en la tecnología *eyetracking*. Se ha realizado una investigación para seleccionar aquellos mecanismos de autenticación que eran idóneos para formar parte de este proyecto.

En primer lugar, ha sido necesario averiguar qué herramientas eran necesarias para diseñar los mecanismos de autenticación. Para superar este primer obstáculo, se ha tenido que realizar una investigación sobre los requerimientos del sistema que se iba a crear. Teniendo en cuenta las habilidades de programación del autor, se ha tomado la decisión de utilizar Java como el lenguaje de programación. Esto dio lugar a la necesidad de documentarse en las librerías de Java que se iban a utilizar y también aprender la API del dispositivo que facilitará el trabajo de desarrollo del código.

Una vez que la implementación de los mecanismos de autenticación y las pruebas han sido realizadas, se ha podido apreciar que la aplicación ha funcionado de la manera esperada. Esto significa que los sistemas implementados y probados demuestran ser una alternativa real a los mecanismos de autenticación tradicionales. No solo proporcionan un nivel de seguridad mayor, sino que también la usabilidad está un paso por delante de los mecanismos de autenticación actuales.

Como se puede ver en los resultados del capítulo cinco, se puede afirmar que se ha conseguido el objetivo del proyecto, ya que la implementación de los mecanismos fue satisfactoria y la comparación entre ellos demuestra que el rendimiento en la mayoría de los casos es aceptable.

Cuando el proyecto se ha terminado y el conocimiento sobre la tecnología *eyetracking* ha sido adquirido en profundidad, se han observado posibles mejoras de las soluciones diseñadas.

6.2 Líneas futuras

Para terminar este Trabajo de Fin de Grado, se ha creado una lista de posibles mejoras futuras que han surgido a raíz de este proyecto:

- **Añadir nuevos algoritmos.** Desarrollar nuevos algoritmos que se puedan sumar a la comparación de los que ya han sido implementados.
    - Lectura de textos [32]
    - *Object PassTiles* [33]
    - *Image PassTiles* [33]
- **Optimización de los algoritmos que han sido desarrollados.** Desarrollo de bloques de código que mejoren la usabilidad y la seguridad de los mecanismos de autenticación implementados.
    - Almacenar las contraseñas en una base de datos
    - Cifrar la contraseña
    - Mejorar la gestión de los puntos mirados
- **Implementar en otros sistemas operativos.** Desarrollo del proyecto en otro sistema operativo compatible con el SDK como puede ser Linux, Mac o Android.
- **Utilizar diferentes idiomas**. El contenido gráfico de las escenas creadas para este proyecto se ha realizado en inglés, pero para incrementar el uso de la aplicación, puede ser una buena idea la traducción del texto a otros idiomas haciendo que sea más accesible para todos.
- **Llevar a cabo el estudio en varios usuarios**. Incluyendo un estudio más profundo en la usabilidad.
- **Añadir más parámetros a la comparación.** Condiciones ambientales (luz, posición, distancia…).